# 10. Rekurentní vztahy

Kapitolu uvedeme populárním příkladem.

**Příklad 10.a:** Tento problém je znám po názvem Hanojské věže. Představte si tři tyčky, na jedné je navlečeno $n$ disků (s dírkou uprostřed) pěkně podle velikosti od největšího dole po nejmenší nahoře. (Chtěl jsem udělat obrázek, ale místo toho vás pošlu do nejbližšího hračkářství, kde v oddělení pro mrňata určitě tyčku s kolečky mají.) Cílem je dostat tyto disky do stejné pozice, ale na jiné tyčce, přičemž jediný povolený tah je přesunout právě jeden disk buď na prázdnou tyč nebo na disk na jiné tyči, který je ale větší; jinými slovy, nelze nosit víc disků najednou a nelze položit větší na menší (a nelze je odložit někde úplně mimo). Je možné tuto úlohu vyřešit?

Ať už vymyslíme jakýkoliv způsob, nakonec musí přijít okamžik, kdy přesouváme dolní, největší disk na cílovou tyč. Abychom to mohli udělat, je nutné všechny disky nad ním dát někam jinam, ale žádný z nich nesmí přijít na cílovou tyč, to bychom totiž ten největší nemohli dát na něj, konec konců, my ten největší stejně chceme dát až dolů. Vidíme tedy, že nutnou přípravou pro přesunutí největšího disku je, aby všechny ostatní byly na té třetí tyči, a to samozřejmě podle velikosti (jinak to nejde). Shrnuto, chceme-li přenést celou hromadu řekněme na tyč 2, musíme tam dát dolů největší disk, což vyžaduje přenesení pyramidy disků nad ním na tyč číslo 3.

Dostáváme tím jasnou rekurzi. Začneme s pyramidou $n$ disků na tyči 1, a abychom ji přenesli na tyč 2, musíme nejprve přenést horních $n-1$ disků na tyč 3. Tuto menší pyramidku o $n-1$ discích přeneseme tak, že její největší disk chceme přenést na cílovou tyč 3, ale na to potřebujeme to, co je nad ním, tedy pyramidku velikosti $n-2$, přenést na tyč 1 či 2 atd. Dříve či později dojdeme k tomu, že máme někam přenést jeden disk, a to ten nejmenší, což lze bez problémů.

Proveditelnost by tedy mělo jít dokázat indukcí. Jediná trochu nejasná věc je, že uprostřed řešení budeme v situaci, kdy máme přenést řekněme pyramidku s 13 disky, ale dalších 5 disků z předchozího rekurzivního rozkladu už se někde potuluje. Nedojde při pokusu o skutečnou realizaci našeho algoritmu ke konfliktu s pravidly? Naštěstí ne. Všechny ty disky z předchozího rozkladu jsou totiž větší než ty v naší pyramidce, tudíž je můžeme v dané chvíli považovat za podlahu, v přesouvání té pyramidky nás neomezí. Raději to zapracujeme do našeho důkazu.

Zkusíme tedy dokázat indukcí, že dokážeme přenést pyramidku $n$ disků z libovolné tyče $a$ na libovolnou jinou tyč $b$, s tím, že na tyčích $b$ a $c$ již mohou být nějaké větší disky.

(0) $n = 1$: Jeden disk určitě přeneseme na cílovou tyč, přičemž nám nebude vadit, když už tam bude nějaký větší disk.

(1) Předpokládáme, že pyramidku o velikosti $n$ umíme. Mějme pyramidku o $n+1$ discích na tyči $a$, potřebujeme ji dostat na tyč $b$, přičemž na tyčích $b$ a $c$ už jsou třeba nějaké disky větší než ty v naší pyramidce. Nejprve použijeme indukční předpoklad a přesuneme horních $n$ disků na tyč $c$ (v tom nám případný větší disk dole nebude vadit), pak přesuneme spodní disk naší pyramidky na tyč $b$ (ani v tom nám případný větší disk nebude vadit), načež opět využijeme indukční předpoklad a přesuneme horních $n$ disků naší pyramidy z tyče $c$ na tyč $b$, kde už leží disk s číslem $n+1$, který je větší než disky pyramidky nad ním, i to je v pořádku.

Tím je důkaz hotov.

To bylo snadné. Mnohem zajímavější je otázka, kolik přesunů disků („tahů“) na to budeme potřebovat. Když označíme $H_n$ počet tahů, které náš algoritmus spotřebuje na přesun pyramidky o $n$ discích, tak nám zkušenosti z kapitoly o indukci naznačují, že pro $H_n$ dostaneme rekurzivní vztah. Je jasné, že $H_1 = 1$. Postup v kroku (1) pak říká, že $H_{n+1} = H_n + 1 + H_n = 2H_n + 1$.

Podobné funkce jsme zkoumali v kapitole 5b. Nejprve jsme si vždy spočítali několik prvních hodnot a pak z toho uhádli vzorec. Zde máme $H_1 = 1$, $H_2 = 3$, $H_3 = 7$, $H_4 = 15$, $H_5 = 31$. Vidíte nějaký vzorec? Pokud ne, zkusíme si ještě jiný přístup, který na to jde z opačné strany, použijeme k nalezení $H_n$ naši rekurzi, trochu optimismu a Větu 9c.2.

$$\begin{aligned} H_n &= 2H_{n-1} + 1 = 2(2H_{n-2} + 1) + 1 = 2^2 H_{n-2} + 2 + 1 \\ &= 2^2(2H_{n-3} + 1) + 2 + 1 = 2^3 H_{n-3} + 2^2 + 2 + 1 \\ &= 2^3(2H_{n-4} + 1) + 2^2 + 2 + 1 = 2^4 H_{n-4} + 2^3 + 2^2 + 2 + 1 = \dots \\ &= 2^{n-1} H_1 + 2^{n-2} + \cdots + 2^2 + 2 + 1 = \sum_{i=0}^{n-1} 2^i = \frac{1-2^n}{1-2} = 2^n - 1. \end{aligned}$$

Ta část se třemi tečkami je samozřejmě podezřelá, to byl ten optimismus. Proto jsme zatím nedokázali, že máme správnou odpověď, ale máme už rozumného kandidáta, pro kterého správnost dokážeme snadno indukcí:

(0) $H_1 = 2^1 - 1 = 1$, to souhlasí.

(1) Předpokládejme, že pro nějaké $n \geq 1$ máme $H_n = 2^n - 1$. Pak

$$H_{n+1} = 2H_n + 1 = 2 \cdot (2^n - 1) + 1 = 2^{n+1} - 1.$$

Souhlasí, náš vzorec je správný.

Řešení hádanky zvané Hanojské věže pro $n$ disků tedy zabere $2^n - 1$ přesunů, pokud použijeme náš algoritmus. Zajímavé je, že lze dokázat (to už je těžší), že hádanku nelze vyřešit za méně tahů, náš algoritmus je tedy optimální řešení.

K této úloze se váže legenda o jakémsi klášteře ve Vietnamu, kde mniši zkoušejí už několik staletí vyřešit tuto úlohu s 64 disky (samozřejmě zlatými a velkými), a až to udělají, tak svět skončí nebo něco. Pokud následují optimální strategii a jeden disk přenesou za sekundu (což je hodně optimistické, jak asi sami znáte, když někdy doma přenášíte velké kusy zlata), tak jim to zabere $2^{64} - 1 \sim 18 \cdot 10^{18}$ vteřin, což vychází na nějakých 600 miliard let. Zatím tedy asi nemá smysl rozfofrovat penzijní fond.

S tím souvisí ještě jedna zajímavost. Jak by vlastně mniši ten ideální rekurzivní algoritmus dělali? Museli by si zapisovat do svitků, ve které fázi kterého podprogramu zrovna jsou a kam se mají vracet, to nevypadá moc prakticky. Naštěstí existuje jednoduchý návod.

Algoritmus: V prvním kroku vezměte nejmenší disk a přemístěte jej směrem doprava na nejbližší tyč, kam vám to pravidla dovolí, přičemž se to bere cyklicky (z poslední tyče se vracíte na první).

V druhém kroku vezměte takový jiný než nejmenší disk, kterým je možné v rámci pravidel táhnout, a také jej přesuňte doprava (cyklicky) na nejbližší možnou tyč.

Tyto dva kroky teď opakujte, v druhém kroku bude vždy jen jediný disk jiný než nejmenší, který je možné přesunout jinam.

Pokud je $n$ liché, celá pyramidka se přesune o jedno doprava, pokud je $n$ sudé, přesune se o jedno doleva (cyklicky). Chcete-li to naopak, šoupejte disky na opačnou stranu.

Zkusíme si to se čtyřmi disky:

$$
\begin{array}{ccc} \mathbf{1}&&\\ 2&&\\ 3&&\\ 4&\perp&\perp \end{array}
\Longrightarrow
\begin{array}{ccc} \mathbf{2}&&\\ 3&&\\ 4&1&\perp \end{array}
\Longrightarrow
\begin{array}{ccc} 3&&\\ 4&\mathbf{1}&2 \end{array}
\Longrightarrow
\begin{array}{ccc} \mathbf{3}&&1\\ 4&\perp&2 \end{array}
\Longrightarrow
\begin{array}{ccc} &&\mathbf{1}\\ 4&3&2 \end{array}
\Longrightarrow
\begin{array}{ccc} 1&&\\ 4&3&\mathbf{2} \end{array}
$$

$$
\Longrightarrow
\begin{array}{ccc} \mathbf{1}&2&\\ 4&3&\perp \end{array}
\Longrightarrow
\begin{array}{ccc} &1&\\ &2&\\ \mathbf{4}&3&\perp \end{array}
\Longrightarrow
\begin{array}{ccc} &\mathbf{1}&\\ &2&\\ \perp&3&4 \end{array}
\Longrightarrow
\begin{array}{ccc} &\mathbf{2}&1\\ \perp&3&4 \end{array}
\Longrightarrow
\begin{array}{ccc} &&\mathbf{1}\\ 2&3&4 \end{array}
$$

$$
\Longrightarrow
\begin{array}{ccc} 1&&\\ 2&\mathbf{3}&4 \end{array}
\Longrightarrow
\begin{array}{ccc} \mathbf{1}&&3\\ 2&\perp&4 \end{array}
\Longrightarrow
\begin{array}{ccc} &&3\\ \mathbf{2}&1&4 \end{array}
\Longrightarrow
\begin{array}{ccc} &&2\\ &&3\\ \perp&\mathbf{1}&4 \end{array}
\Longrightarrow
\begin{array}{ccc} &&1\\ &&2\\ &&3\\ \perp&\perp&4 \end{array}
$$

Je vidět, že v každém kroku, kdy nenosíme disk 1, máme opravdu jen jednu jinou možnost tahu.

Mimochodem, zkoumají se i varianty tohoto problému. Jedna z nich (Reve's puzzle) je, že ty tyčky jsou čtyři, a zajímavé na ní je, že se dodnes neví, kolik je minimum tahů k přemístění $n$ disků. Kandidátem je algoritmus z roku 1939, ale zatím (2011) se nepovedlo dokázat, že je to optimální strategie.

Jako doplňkové čtení silně doporučuji povídku A.C. Clarka *Devět miliard božích jmen.*

$\triangle$

Induktivně definované funkce či posloupnosti dostáváme, když se při popisu situace setkáváme se vztahem, který nějak kombinuje přítomnost s minulostí. Takový vztah má své jméno.

**!**

**Definice**

**Rekurentní vztah** či **rekurzivní vztah** (**recurrent relation**) pro posloupnost $\{a_k\}$ je libovolná rovnice typu $F(a_n, a_{n-1}, a_{n-2}, \dots, a_{n_0}) = 0$, kde $F$ je nějaká funkce.

**!** Třeba podstata problému Hanojských věží se dá vyjádřit vztahem $H_n - 2H_{n-1} - 1 = 0$. Je to vztah výjimečně pěkný, rozhodně lepší než třeba vztah $a_n = \sin(a_{n-1}a_{n+1})\sqrt{\frac{a_{n-2}^4+1}{a_{13}+7}}$, se kterým bychom upřímně řečeno nehnuli.

Hlavním problémem rekurentních vztahů je, že nenabízí rozumný způsob, jak počítat neznámé hodnoty. Například v rekurentním vztahu $a_n^8 - a_n a_{n-1} + a_{n-2} = 0$ nám příliš nepomůže, když známe třeba $a_1 = a_2 = 1$, protože rovnici $a_3^8 - a_3 + 1 = 0$ vyřešit neumíme.

Není proto divu, že se většina pojednání o rekurentních vztazích hned na začátku omezí jen na ty vztahy, které lze upravit do tvaru $a_n = G(a_{n-1}, a_{n-2}, \dots, a_{n_0})$. Často se takovýto vztah píše s posunutým indexem, $a_{n+1} = G(a_n, a_{n-1}, \dots, a_{n_0})$, to „$n + 1$“ psychologicky naznačuje, že něco máme ($a_{n_0}$ až $a_n$) a chceme další člen. Takto jsme to dělávali v kapitole 5. Vztah tohoto typu je mnohem perspektivnější, protože jakmile známe několik počátečních hodnot, dokážeme iterací počítat nové a nové hodnoty. Tím ale narážíme na další problém. Výpočet konkrétních hodnot tímto způsobem je velice drahý, na $a_{10000}$ potřebujeme spočítat všechna předchozí $a_n$.

Hlavním tématem proto bývá hledání způsobu, jak pro rekurentně zadanou posloupnost (či funkci) najít explicitní vyjádření pro $a_n$ vzorcem v uzavřeném tvaru (elementární funkce spojené algebraickými operacemi či skládáním), viz ten příklad s věžemi. Někdy je to z principu nemožné, někdy by to snad možné i bylo, ale neumíme to najít. Abychom dostali rozumné odpovědi, budeme se muset omezit na rekurentní vztahy velice jednoduchého (ale pořád silně užitečného) typu. Pro ilustraci ukážeme ještě několik příkladů.

**Příklad 10.b:** Připomeňme si příklad 5a.d, kde jsme rekurzí dokázali tapetovatelnost šachovnice triminy. Otevřená otázka zůstala, kolik trimin je na šachovnici o straně $2^n$ potřeba, označili jsme to $t_n$. Algoritmus vedl na rovnice $t_1 = 1$ a $t_{n+1} = 4t_n + 1$. Zkusíme postup z hanojského příkladu 10.a.

$$\begin{aligned} t_n &= 4t_{n-1} + 1 = 4(4t_{n-2} + 1) + 1 = 4^2 t_{n-2} + 4 + 1 \\ &= 4^2(4t_{n-3} + 1) + 4 + 1 = 4^3 t_{n-3} + 4^2 + 4 + 1 \\ &= 4^3(4t_{n-4} + 1) + 4^2 + 4 + 1 = 4^4 t_{n-4} + 4^3 + 4^2 + 4 + 1 = \dots \\ &= 4^{n-1} t_1 + 4^{n-2} + \dots + 4^2 + 4 + 1 = \sum_{i=0}^{n-1} 4^i = \frac{1-4^n}{1-4} = \tfrac{1}{3}(4^n - 1). \end{aligned}$$

Ověření správnosti tohoto vzorce indukcí necháme na čtenáři.
△

Vidíme, že jsme schopni tímto postupem relativně rychle nalézat vzorce pro funkce či posloupnosti zadané vztahem $a_{n+1} = a \cdot a_n + b$ a počáteční hodnotou $a_{n_0} = A$. Bohužel pro komplikovanější vztahy už to není příliš perspektivní.

**Příklad 10.c:** Mějme induktivní definici funkce $f(1) = 3$, $f(2) = -1$ a $f(n) = f(n-1) + 6f(n-2)$ pro $n \geq 3$. Dostáváme $f(3) = 17$, $f(4) = 11$, $f(5) = 113$, $f(6) = 209$. Vidíte v tom nějaký vzorec? Já ne.

Zkusme přístup z předchozích příkladů:

$$\begin{aligned} f(n) &= f(n-1) + 6f(n-2) = [f(n-2) + 6f(n-3)] + 6f(n-2) = 7f(n-2) + 6f(n-3) \\ &= 7[f(n-3) + 6f(n-4)] + 6f(n-3) = 13f(n-3) + 42f(n-4) \\ &= 13[f(n-4) + 6f(n-5)] + 42f(n-4) = 55f(n-4) + 78f(n-5) = ? \end{aligned}$$

Vidíte z toho něco? Asi bude lepší počkat na kapitolu 10b, kde se podobné příklady naučíme řešit na dvou řádcích.
△

## 10a. Lineární rekurentní rovnice

Hodně rekurentních vztahů se dá přepsat do tvaru, kdy $G$ závisí jen na stále stejném počtu $k$ předchozích členů a navíc lineárním způsobem: $a_n = d_1(n)a_{n-1} + d_2(n)a_{n-2} + \dots + d_k(n)a_{n-k}$.

Z praktického důvodu bude lepší takovéto rovnice psát trochu jinak.

**!**

**Definice**

**Lineární rekurentní rovnice**, popřípadě **lineární rekurzivní rovnice řádu** $k \in \mathbb{N}_0$ je libovolná rovnice ve tvaru

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \dots + c_2(n)a_{n+2} + c_1(n)a_{n+1} + c_0(n)a_n = b_n \qquad \text{pro všechna } n \geq n_0,$$

kde $n_0 \in \mathbb{Z}$, $c_i(n)$ pro $i = \{0, \dots, k-1\}$ (tzv. **koeficienty** rovnice) jsou nějaké funkce $\mathbb{Z} \mapsto \mathbb{R}$, přičemž $c_0(n)$ není identicky nulová funkce, a $\{b_n\}_{n=n_0}^{\infty}$ (tzv. **pravá strana rovnice**) je pevně zvolená posloupnost reálných čísel.

Jestliže $b_n = 0$ pro všechna $n \geq n_0$, pak se příslušná rovnice nazývá **homogenní**.

By a **linear recurrence equation** of order $k \in \mathbb{N}_0$ we mean any equation of the form

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \dots + c_2(n)a_{n+2} + c_1(n)a_{n+1} + c_0(n)a_n = b_n \qquad \text{for all } n \geq n_0,$$

where $n_0 \in \mathbb{Z}$, $c_i(n)$ for $i = \{0, \dots, k-1\}$ (**coefficients** of the equation) are some functions $\mathbb{Z} \mapsto \mathbb{R}$ with $c_0(n)$ not identically zero, and $\{b_n\}_{n=n_0}^{\infty}$ (the **right hand-side** of the equation) is a fixed sequence of real numbers.

If $b_n = 0$ for all $n \geq n_0$, then the equation is called **homogeneous**.

Příklady ze začátku této kapitoly sem samozřejmě patří, stačí ve vztazích vhodně posunout indexy. Hanojský vztah se dá přepsat jako $H_{n+1} - 2H_n = 1$, jde o lineární rekurentní rovnici 1. řádu, podobně se dá triminová rovnice přepsat jako $t_{n+1} - 4t_n = 1$. Ve třetím příkladě jsme měli rovnici $f(n+2) - f(n+1) - 6f(n) = 0$, je to tedy lineární rekurentní rovnice 2. řádu. Naopak $a_n - a_{n-1}a_{n-2} = 0$ lineární není.

**Poznámka:** Máme-li posloupnost $\{a_n\}$, lze vytvořit z rozdílu následujících členů posloupnost novou, definovanou jako $\Delta a_n = a_n - a_{n-1}$. Tomuto se říká *diference* $\{a_n\}$ a existují úlohy, které s tímto pojmem pracují a vedou na rovnice typu $a_{n+1} = \Delta a_n + 1$ a podobně. Těmto rovnicím se přirozeně říká diferenční rovnice. Někteří autoři pak tento název přenášejí na všechny rekurentní rovnice a mluví o lineárních diferenčních rovnicích.
△

**! 10a.1 Poznámka o rovnicích a posunu indexu:**

Stojí za zmínku, že vlastně nejde o rovnici jednu, ale o nekonečně mnoho rovnic, například ten hanojský vztah vlastně znamená rovnice $H_2 - 2H_1 = 1$, $H_3 - 2H_2 = 1$, $H_4 - 2H_3 = 1$, $H_5 - 2H_4 = 1$ a tak dále. Je to triviální, ale až budeme mluvit o „řešení rovnice“, je dobré si toho být vědom, pod slovem „rovnice“ se jich skrývá mnoho.

S tím souvisí posouvání indexu. Viděli jsme již tři formální způsoby, jak rovnice napsat. V příkladě 10.c jsme použili zápis $f_n = f_{n-1} + 6f_{n-2}$ typu $f_n = G(f_{n-1}, f_{n-2})$, zmínili jsme také intuitivnější formu $f_{n+1} = f_n + 6f_{n-1}$ neboli $f_{n+1} = G(f_n, f_{n-1})$. Definice lineárních rekurentních rovnic teď po nás chce zapsat tento vztah jako $f_{n+2} - f_{n+1} - 6f_n = 0$, kdy jsme to „referenční $n$“ dali naopak ke členu s nejmenším indexem ve vztahu.

Nicméně toto není pro lineární rovnice standard, různí autoři si své lineární rovnice indexují všelijak podle toho, jak jim to přijde výhodnější. K tomu se obvykle dodává, že v praxi není problém přecházet mezi jednotlivými formami pomocí posunu indexu. Co z toho pro nás plyne?

Když otevřeme nějakou knihu o takovýchto rovnicích, tak se musíme dobře podívat, jak zrovna dotyčný autor rovnice chce mít, abychom správně chápali jeho tvrzení. Nepříjemné je to u teorie, kdy je třeba při přechodu z jedné knihy do druhé překládat vzorce do jiného značení.

Je také jasné, že se musíme naučit dobře měnit index. Při sestavování rovnic pro popis nějaké konkrétní situace je totiž výhodné pracovat s tím „psychologickým“ tvarem $a_{n+1} = \dots$, pro řešení jej pak budeme přepisovat do tvaru dle definice lineární rovnice. Jak tedy indexy posouváme?

Na první pohled snadno, prostě všechny výskyty indexační proměnné zvýšíme či snížíme o stejné číslo. U dané rovnice $f_n = f_{n-1} + 6f_{n-2}$ přičtením dvojky ke všem $n$ dostáváme $f_{n+2} = f_{n+1} + 6f_n$ neboli $f_{n+2} - f_{n+1} - 6f_n = 0$, přesně jak potřebujeme pro tuto knihu. Je přitom ale třeba mít na paměti dvě věci. Ukážeme si je na rovnici $a_{n+1} = a_n - n^2 a_{n-3} + 2n$ pro $n \geq 7$, kde potřebujeme zvýšit index o tři.

Za prvé, když se mění indexační proměnná, mění se opravdu všude, nejen v místech indexu. Pokud bychom naši vzorovou rovnici přepsali jako $a_{n+4} = a_{n+3} - n^2 a_n + 2n$, bylo by to špatně, viz $n^2$ a $2n$. Správný převod je $a_{n+4} = a_{n+3} - (n+3)^2 a_n + 2(n+3)$. Opomenutí takového posunu je častá chyba u zkoušek.

Za druhé, indexační proměnná se mění i ve specifikaci rozsahu. To právě souvisí s tím, že vlastně mluvíme o soustavě rovnic. Původní zadání $a_{n+1} = a_n - n^2 a_{n-3} + 2n$ pro $n \geq 7$ obsahovalo tyto rovnice:

$$
\begin{aligned}
a_8 &= a_7 - 7^2 a_4 + 14\\
a_9 &= a_8 - 8^2 a_5 + 16\\
a_{10} &= a_9 - 9^2 a_6 + 18\\
a_{11} &= a_{10} - 10^2 a_7 + 20 \quad \text{atd.}
\end{aligned}
$$

My musíme dosáhnout toho, aby rovnice po posunu indexu dávala stejnou množinu. Na to doporučíme dva způsoby.

Jedna možnost je spolehnout se na selský rozum. Ze zadání „$a_{n+1} = a_n - n^2 a_{n-3} + 2n$ pro $n \geq 7$“ vidíme, že nejmenší možný index, který se v rovnicích může vyskytnout, je $7 - 3 = 4$. Musíme zajistit, aby tomu tak bylo i v naší přeindexované rovnici, což se evidentně udělá volbou $n \geq 4$. Ověříme, po dosazení $n = 4$ do přepsané rovnice pak první rovnost vychází $a_8 = a_7 - 7^2 a_4 + 14$, což je správně, viz ten sloupeček výše. Řešíme tedy rovnici $a_{n+4} - a_{n+3} + (n+3)^2 a_n = 2n + 6$ pro $n \geq 4$, je to lineární rekurentní rovnice 4. řádu.

Další možnost je při přečíslování rovnic použít formální substituci, viz příklad 10b.f.
△

Když už jsme u zápisu, co jsou ty tři tečky v definici rekurentní rovnice? V matematice se rozumí, že takto definované výrazy v sobě ve skutečnosti schovávají indukci, v tomto případě se dá vyjádřit pomocí sumačního znaménka:

$$a_{n+k} + \sum_{i=0}^{k-1} c_i(n) a_{n+i} = b_n.$$

Trochu nevýhoda takového značení je, že si jej člověk musí zase v hlavě překládat do dlouhých součtů, zejména chce-li to použít v konkrétní situaci, takže z důvodu názornosti se budeme v kritických chvílích snažit používat spíš to delší značení. Na druhou stranu je v tomto sumačním značení pěkně vidět základní referenční index $n$ a řád $k$, navíc je to podstatně kratší na psaní, takže v teoretických výpočtech (zejména těch brutálnějších) se sumační zápis vyloženě vyplatí.

Dodejme ještě, že ta podmínka $c_0 \neq 0$ je podstatná pro určení řádu rovnice, jinak by tam nula nevadila. Například rekurentní rovnici 1. řádu $a_{n+1} - 13a_n = 0$ lze také psát jako $a_{n+3} - 13a_{n+2} = 0$, což vlastně odpovídá obecnému tvaru $a_{n+3} + c_2 a_{n+2} + c_1 a_{n+1} + c_0 a_n = 0$, ale zde $c_0 = 0$, a tak to rovnice 3. řádu není. V principu by šlo používat i tu druhou variantu (pokud tomu člověk dobře rozumí), ale my se zde budeme držet „správného“ zápisu.

Abychom tu podmínku, že $c_0(n)$ není nulová funkce, nemuseli pořád psát, tak namísto toho řekneme, že daná rovnice je řádu $k$, a bude to totéž. Jsou ale situace, kde řád tak podstatný není, pak předpoklad o řádu rovnice psát nebudeme.

V této kapitole teoreticky prozkoumáme, jak se řešení lineárních rekurentních rovnic chovají, konkrétní metody pro jejich hledání pak odvodíme v další kapitole. Nejprve bychom si měli říct, co vlastně při řešení rovnic hledáme.

! **Definice**
Nechť je dána lineární rekurentní rovnice

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \cdots + c_1(n)a_{n+1} + c_0(n)a_n = b_n \quad \text{pro všechna } n \geq n_0.$$

Jako její **řešení** označíme libovolnou posloupnost $\{a_n\}_{n=n_0}^{\infty}$ takovou, že po dosazení odpovídajících členů do dané rovnice dostáváme pro všechna $n$ pravdivý výrok.

Ukážeme jednoduchý příklad.

**Příklad 10a.a:** Posloupnost $a_n = 13(n-1)!$, $n \geq 1$ je řešením rovnice

$$a_{n+1} - na_n = 0 \quad \text{pro všechna } n \geq 1,$$

protože když pro libovolné $n \in \mathbb{N}$ dosadíme dotyčný vzorec do rovnice za $a_n$ a $a_{n+1}$, tak dostaneme pravdivý výrok $13n! - n \cdot 13(n-1)! = 0$.
△

Mimochodem, tato rovnice už je dost těžší a zde se takové řešit nenaučíme.

! **Příklad 10a.b:** Dlouhodobé pozorování volně žijících králíků v Austrálii ukázalo, že se jejich počet každých šest měsíců zdvojnásobí. Vše začalo v roce 1859, kdy jich bylo vypuštěno 24. Kolik jich bylo v roce 1869?

Označme si jako $a_n$ počet králíků v roce $1859 + n$, takže nás zajímá $a_n$ pro $n \geq 0$ a víme, že $a_0 = 24$. Ze vstupních dat vyplývá lineární rekurentní rovnice, kterou se růst králíků řídí: Je-li jich jeden rok $a_n$, pak za šest měsíců je jich $2a_n$ a za dalších šest měsíců (tedy za rok od $a_n$) je jich $2 \cdot 2a_n = 4a_n$. Dostáváme $a_{n+1} = 4a_n$, tedy $a_{n+1} - 4a_n = 0$ pro $n \geq 0$.

Hledáme řešení této rovnice 1. řádu, zároveň chceme, aby splňovalo $a_0 = 24$. Spočítáme začátek této posloupnosti:

$$\begin{aligned}
a_0 &= 24,\\
a_1 &= 4a_0 = 4 \cdot 24,\\
a_2 &= 4a_1 = 4 \cdot (4 \cdot 24) = 4^2 \cdot 24,\\
a_3 &= 4a_2 = 4 \cdot (4^2 \cdot 24) = 4^3 \cdot 24.
\end{aligned}$$

Odhadneme, že posloupnost $a_n = 24 \cdot 4^n$ je hledanou posloupností.

Zkouška: $a_0 = 24 \cdot 4^0 = 24$, počáteční hodnota je v pořádku.

Pro libovolné $n \geq 0$ pak $a_{n+1} - 4a_n = 24 \cdot 4^{n+1} - 4 \cdot 24 \cdot 4^n = 0$, přesně jak bylo požadováno.

Pak už hravě zjistíme, že v roce 1869 tam teoreticky bylo $a_{10} = 24 \cdot 4^{10}$ neboli cca 25 milionů králíků. To už je skoro dost, není někde chyba? Historie říká, že od roku 1869 zabíjeli v Austrálii asi 2 milony králíků ročně, aniž by to nějak znatelněji ovlivnilo jejich populaci, takže těch 24 milionů najednou nevypadá nereálně. Pokusy o klasickou decimaci (odstřel, jedy) růst mírně zpomalily, ale i tak byl v roce 1950 jejich stav odhadován na 600 milionů. Moderní metody s tím drobet pohly, dnes je stav odhadován na 300 milionů, což je ale pořád slušné číslo na 24 prapředků.

Pokročilá poznámka: Čtenář asi cítí, že tento model není zrovna nejspolehlivější. Zádrhel je ve výchozím předpokladu, populace se za šest měsíců bude těžko přesně zdvojnásobovat, spíš půjde o průměrnou hodnotu. Jde tedy o statistický údaj a statistické údaje se nedají používat na malé populace, fungují jen na velkých množstvích. Kdybychom zkoumali populace o milionech králíků (což později děláme), tak by ještě výsledky mohly být docela spolehlivé, ale my jsme začali s 24 králíky, což je hodně málo. U tak malé populace se klidně může stát, že za šest měsíců jich nebude $2 \cdot 24 = 48$, ale třeba 34 či 50, čímž se celý další vývoj vykolejí dost jinam. Při modelování reálných situací je vždy třeba být opatrný, zda výsledek, který nám matematika dala, lze aplikovat na popisovanou situaci. Klíčová je zde věrnost modelu, matematická odpověď na matematickou otázku je samozřejmě spolehlivá.

Ke králíkům se vrátíme v příkladě 10b.i, kde se podíváme za rok 1869, a také v příkladě 10b.b, kde ukážeme jiný přístup k jejich problematice.
△

**!** Vraťme se k příkladu o králících. Když si místo počátečního počtu 24 dáme počet jiný, třeba obecně číslo $c$, pak stejným postupem dostaneme řešení $a_n = c \cdot 4^n$. Jinak řečeno, pokud uvažujeme jen danou rovnici $a_{n+1} = 4a_n$, pak ji řeší všechny posloupnosti typu $a_n = c \cdot 4^n$, kde $c$ je nějaká reálná konstanta (tohle si zase ověřte dosazením). Máme tedy nekonečně mnoho řešení daných jedním vzorečkem, ve kterém si můžeme úpravou parametru vybírat jedno konkrétní řešení, které vyhovuje doplňkovým požadavkům.

Z hlediska filosofického to dává smysl. Rovnice samotná je něco jako přírodní zákon popisující množení králíků, tudíž se dá čekat, že existuje mnoho situací, které tomuto zákonu vyhovují, stejně jako gravitační zákon umožňuje mnoho různých způsobů padání objektů (kladivo padající ze střechy mrakodrapu urazí trochu jinou dráhu než chleba padající ze stolu). Takovéto situace se liší svobodou, v našem případě králíků lze mluvit o situaci s jedním stupněm volnosti, což vidíme podle jednoho parametru. Logicky pak do toho zapadá, že když si přidáme jeden další požadavek, tak už tuto volnost ztratíme a dostaneme jedno konkrétní řešení.

Brzy ukážeme, že když je rovnice řádu $k$, tak má $k$ stupňů volnosti, neboli lze najít řešení, ve kterém je $k$ konstant, které si můžeme zvolit dle libosti. Takovému řešení se říká **obecné řešení**.

Pokud začneme klást požadavky, tak s každým požadavkem jeden stupeň volnosti ubyde (pokud je klademe rozumně, aby si třeba neodporovaly nebo některé z nich neříkaly totéž). Pokud si rozumně zadáme $k$ podmínek, tak už zbyde jen jedno řešení. Každému takovému konkrétnímu řešení říkáme **partikulární řešení**. Pokud bychom si například u těch králíků zadali, že $a_5 = 40000$, tak se najde přesně jedno partikulární řešení, které tomu bude vyhovovat, a my z něj vyčteme, s kolika králíky bychom měli začít, aby z nich za 5 let bylo 40000.

Máme-li rovnici druhého řádu, pak bude mít obecné řešení dva parametry, a pokud si zadáme třeba $a_0 = 1$ a $a_{10} = 13$, tak už tím určíme řešení jednoznačně. Podmínky si tedy můžeme zadávat všelijak, ale nejčastěji to děláme tak, jak jsme zvyklí z kapitoly 5b: Řekneme, jak má hledaná posloupnost začít.

Například rovnici $a_{n+2} - 2a_{n+1} + a_n = n + 1$ řádu 2 si můžeme přepsat (po posunu indexu $n + 1 \mapsto n$) do intuitivního tvaru $a_{n+1} = 2a_n - a_{n-1} + n$. Když si zadáme, kolik je $a_0$ a $a_1$, tak už se ostatní hodnoty dají dopočítat neboli rovnice má jediné řešení.

Této myšlence dáme oficiální název.

**!** **Definice**
Nechť je dána lineární rekurentní rovnice řádu $k$

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \cdots + c_1(n)a_{n+1} + c_0(n)a_n = b_n \quad \text{pro všechna } n \geq n_0.$$

Za **počáteční podmínky** (**initial conditions**) pro tuto rovnici považujeme libovolnou soustavu rovnic $a_{n_0} = A_0$, $a_{n_0+1} = A_1$, … , $a_{n_0+k-1} = A_{k-1}$, kde $A_i \in \mathbb{R}$ jsou pevně zvolená čísla.

Takže pro rovnici řádu $k$ definujeme $k$ podmínek, zkušenost pak říká, že řešení nezbyde žádná svoboda. Potvrdí nám to oficiálně následující věta.

**!** **Věta 10a.2.** (o existenci a jednoznačnosti)
Každá lineární rekurentní rovnice má nějaké řešení.
Je-li dána lineární rekurentní rovnice řádu $k$ a příslušné počáteční podmínky, pak existuje jediné řešení této rovnice, které splňuje dané počáteční podmínky.

Věty o existenci a jednoznačnosti bývají v matematických teoriích o rovnicích klíčové. Všimněte si, že druhá část věty vlastně říká dvě věci, jednak že řešení existuje a pak že je jediné. Není to přitom opakování z první části. První část věty totiž jenom říká, že nějaké řešení existuje, ale nezaručí, že nutně musí splňovat počáteční podmínky, které zrovna potřebujeme. Až druhá část říká, že když si libovolně vybereme počáteční podmínky, tak už k nim řešení existuje, a pak také dodá, že je jediné možné.

**Důkaz** (poučný)**:** 1) Dokážeme druhou část věty. Uvažujme rovnici $a_{n+k} + \sum\limits_{i=0}^{k-1} c_i(n)a_{n+i} = b_n$ a počáteční podmínky $a_{n_0} = A_0$, $a_{n_0+1} = A_1$, … , $a_{n_0+k-1} = A_{k-1}$. Nejprve ukážeme, že existuje řešení této úlohy. Sestrojíme jej pomocí strukturální indukce pro $n \in \{n_0, n_0 + 1, n_0 + 2, \dots\}$.

(0) Základní krok: Pro $n = n_0, n_0 + 1, \dots, n_0 + k - 1$ definujeme $a_n = A_{n-n_0}$.

(1) Nechť $n \in \mathbb{N}$, $n \geq n_0 + k - 1$. Předpokládejme, že jsou definovány $a_{n_0}, a_{n_0+1}, \dots, a_n$. Pak definujeme $a_{n+1} = b_{n+1-k} - \sum\limits_{i=0}^{k-1} c_i(n + 1 - k)a_{n+1-k+i}$. Na pravé straně používáme jen $a_{n+1-k}, a_{n+2-k}, \dots, a_n$, které dle

předpokladu máme, neboť díky $n \geq n_0 + k - 1$ všechny používané indexy splňují $n + 1 - k + i \geq n_0$. Definice je proto korektní.

Tím máme vytvořenu posloupnost $\{a_n\}_{n=n_0}^{\infty}$. Potřebujeme ukázat, že je to řešení.

Platnost počátečních podmínek je zjevná dle definice v kroku (0). Nyní ověříme platnost rovnice. Zvolme tedy libovolné $N \geq n_0$. Pak $N + k > n_0 + k - 1$, proto byl při definici $a_{N+k}$ použit krok (1) pro $n = N + k - 1$. Měli jsme tedy definici $a_{N+k} = b_N - \sum\limits_{i=0}^{k-1} c_i(N)a_{N+i}$, což po přepsání dává $a_{N+k} + \sum\limits_{i=0}^{k-1} c_i(N)a_{N+i} = b_N$ a řešená rovnice je splněna.

Nyní je třeba ukázat jendoznačnost. Vezměme tedy jiné řešení $\{\tilde{a}_n\}$ naší úlohy (rovnice a počátečních podmínek). Ukážeme silnou indukcí, že se již musí shodovat s naším $\{a_n\}$. Přesně, pro $n \geq n_0$ dokážeme $V(n)$: $\tilde{a}_n = a_n$.

(0) Pro $n = n_0, n_0 + 1, \dots, n_0 + k - 1$ musí podle počátečních podmínek platit $\tilde{a}_n = A_{n-n_0} = a_n$.

(1) Nechť $n \in \mathbb{N}$, $n \geq n + k - 1$. Předpokládejme, že pro $m = n_0, n_0 + 1, \dots, n$ platí $\tilde{a}_m = a_m$. Protože je $\{\tilde{a}_n\}$ řešení, musí platit $\tilde{a}_{N+k} + \sum\limits_{i=0}^{k-1} c_i(N)\tilde{a}_{N+i} = b_N$. Když to aplikujeme s $N = n + 1 - k$, dostáváme $\tilde{a}_{n+1} + \sum\limits_{i=0}^{k-1} c_i(n + 1 - k)\tilde{a}_{n+1-k+i} = b_{n+1-k}$ neboli $\tilde{a}_{n+1} = b_{n+1-k} - \sum\limits_{i=0}^{k-1} c_i(n + 1 - k)\tilde{a}_{n+1-k+i}$. Použijeme indukční předpoklad a podmínku (1) z definice $\{a_n\}$ a dostáváme

$$\tilde{a}_{n+1} = b_{n+1-k} - \sum_{i=0}^{k-1} c_i(n + 1 - k)\tilde{a}_{n+1-k+i} = b_{n+1-k} - \sum_{i=0}^{k-1} c_i(n + 1 - k)a_{n+1-k+i} = a_{n+1}.$$

Tím je indukční krok dokázán a tedy i jednoznačnost nalezeného řešení.

2) Teď dokážeme první část věty, tedy existenci nějakého řešení libovolné lineární rekurentní rovnice. To je ale snadné. Když je dána lineární rekurentní rovnice řádu $k$, tak si prostě zvolíme nějaké počáteční podmínky, třeba pro jednoduchost $a_{n_0} = a_{n_0+1} = \ldots = a_{n_0+k-1} = 0$, a podle 1) k nim najdeme řešení, máme proto nějaké řešení dané rovnice.

□

**Důsledek 10a.3.**
Nechť $\{a_m\}_{m=n_0}^{\infty}$ a $\{\tilde{a}_m\}_{m=n_0}^{\infty}$ jsou dvě řešení téže lineární rekurentní rovnice řádu $k$. Jestliže se shoduje prvních $k$ členů těchto řešení, pak se shodují celé posloupnosti.

**Důkaz** (poučný)**:** Označme $A_0 = a_{n_0}, A_1 = a_{n_0+1}, \dots, A_{k-1} = a_{n_0+k-1}$. Pak $\{a_m\}$ řeší onu lineární rekurentní rovnici a také splňuje počáteční podmínky $A_i$. Díky shodnosti prvních $k$ členů ovšem i řešení $\{\tilde{a}_m\}$ splňuje tyto počáteční podmínky, proto podle Věty o jednoznačnosti musí jít o shodné posloupnosti.

□

Věta nám zaručila existenci řešení, ale neporadila, jak jej najít vyjádřené explicitním vzorečkem. Abychom tuto otázku dokázali uspokojivě zodpovědět, musíme vědět víc o tom, jakou strukturu řešení mají. Následující věty nepřekvapí nikoho, kdo se již s nějakými lineárními rovnicemi setkal. Jako obvykle ukážeme, že řešení homogenních rovnic Utvoří lineární (vektorový) prostor. Vhodné posuny tohoto prostoru pak dají řešení dané rovnice pro nenulové pravé strany. To je v kostce obsah zbytku této kapitoly, začneme posledně zmíněným faktem.

**!**

**Definice**
Uvažujme lineární rekurentní rovnici

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \cdots + c_2(n)a_{n+2} + c_1(n)a_{n+1} + c_0(n)a_n = b_n \qquad \text{pro všechna } n \geq n_0.$$

Pak se lineární rekurentní rovnice

$$a_{n+k} + c_{k-1}(n)a_{n+k-1} + \cdots + c_2(n)a_{n+2} + c_1(n)a_{n+1} + c_0(n)a_n = 0 \qquad \text{pro všechna } n \geq n_0,$$

nazývá k ní **přidružená homogenní rovnice**.

Přichází první inzerovaná věta.

! **Věta 10a.4.** (o struktuře řešení lineární rekurentní rovnice)
Nechť je dána lineární rekurentní rovnice
$$a_{n+k}+c_{k-1}(n)a_{n+k-1}+\dots+c_1(n)a_{n+1}+c_0(n)a_n=b_n \quad \text{pro všechna } n\ge n_0$$
a nějaké její řešení $\{a_{p,n}\}_{n=n_0}^{\infty}$.
Posloupnost $\{a_n\}_{n=n_0}^{\infty}$ je řešením této rovnice právě tehdy, pokud se dá napsat jako $\{a_n\}=\{a_{p,n}\}+\{a_{h,n}\}$, kde $\{a_{h,n}\}_{n=n_0}^{\infty}$ je nějaké řešení přidružené homogenní rovnice.

! Tento důkaz je v zásadě stejný jako u lineárních rovnic v předchozích kapitolách. Zda je posloupnost řešením dokážeme prostě tak, že ji dosadíme do příslušné rovnice; také víme, že praktičtější je dosadit zkoumanou posloupnost do levé (komplikovanější) strany rovnosti a postupně se úpravami propracovat k pravé.

**Důkaz** (rutinní, poučný)**:** Dokazujeme oba směry ekvivalence.

1) $\Longleftarrow$: Nechť $\{a_{h,n}\}_{n=n_0}^{\infty}$ je nějaké řešení přidružené homogenní rovnice a $\{a_n\}=\{a_{p,n}\}+\{a_{h,n}\}$. Chceme ukázat, že jsme dostali řešení dané rovnice. Dosadíme tedy do rovnice, začneme levou stranou. Pro libovolné $n\ge n_0$ máme
$$\begin{aligned}
&a_{n+k}+c_{k-1}(n)a_{n+k-1}+\dots+c_1(n)a_{n+1}+c_0(n)a_n\\
&\quad=(a_{p,n+k}+a_{n+k})+c_{k-1}(n)(a_{p,n+k-1}+a_{h,n+k-1})+\dots+c_1(n)(a_{p,n+1}+a_{h,n+1})+c_0(n)(a_{p,n}+a_{h,n})\\
&\quad=[a_{p,n+k}+c_{k-1}(n)a_{p,n+k-1}+\dots+c_1(n)a_{p,n+1}+c_0(n)a_{p,n}]\\
&\qquad+[a_{h,n+k}+c_{k-1}(n)a_{h,n+k-1}+\dots+c_1(n)a_{h,n+1}+c_0(n)a_{h,n}]\\
&\quad=b_n+0=b_n.
\end{aligned}$$
Takže $\{a_{p,n}+a_{h,n}\}_{n=n_0}^{\infty}$ řeší danou rovnici.

2) $\Longrightarrow$: Nechť $\{a_n\}_{n=n_0}^{\infty}$ je nějaké řešení dané rovnice. Definujme $a_{h,n}=a_n-a_{p,n}$. Pak evidentně platí rovnost $\{a_n\}=\{a_{p,n}\}+\{a_{h,n}\}$ a zbývá ukázat, že $\{a_{h,n}\}_{n=n_0}^{\infty}$ řeší přidruženou homogenní rovnici. Dosadím do levé strany, použijeme sumační zápis, abychom ukázali, jak se tím věci pěkně zjednoduší.
$$\begin{aligned}
a_{h,n+k}+\sum_{i=0}^{k-1}c_i(n)a_{h,n+i}&=(a_{n+k}-a_{p,n+k})+\sum_{i=0}^{k-1}c_i(n)(a_{n+i}-a_{p,n+i})\\
&=a_{n+k}-a_{p,n+k}+\sum_{i=0}^{k-1}c_i(n)a_{n+i}-\sum_{i=0}^{k-1}c_i(n)a_{p,n+i}\\
&=\Big(a_{n+k}+\sum_{i=0}^{k-1}c_i(n)a_{n+i}\Big)-\Big(a_{p,n+k}+\sum_{i=0}^{k-1}c_i(n)a_{p,n+i}\Big)=b_n-b_n=0.
\end{aligned}$$
□

! Když to zapíšeme množinově, tak množina všech řešení dané lineární rekurentní rovnice je
$$\big\{\{a_{p,n}\}+\{a_{h,n}\};\ \{a_{h,n}\} \text{ řeší přidruženou homogenní rovnici}\big\}.$$
Z toho plyne, že abychom uměli řešit lineární rekurentní rovnice, tak potřebujeme umět opravdu dobře řešit homogenní rovnice, zatímco pro ty nehomogenní stačí nějak uhodnout alespoň jedno řešení. Přesně tuto strategii teď budeme následovat, nejprve se blíže podíváme na to, jak vypadají všechna řešení homogenní rovnice. Na to máme další větu, tentokráte už budeme pracovat jen se sumami, život je krátký.

! **Věta 10a.5.** (o struktuře prostoru řešení homogenní lineární rekurentní rovnice)
Nechť je dána homogenní lineární rekurentní rovnice řádu $k$
$$a_{n+k}+\sum_{i=0}^{k-1}c_i(n)a_{n+i}=0 \quad \text{pro všechna } n\ge n_0.$$
Pak množina $M$ všech jejích řešení je lineární prostor dimenze $k$.

**Důkaz** (drsný, poučný)**:** 1) Nejprve dokážeme, že $M$ je lineární prostor. Vezměme dva prvky $\{a_n\}$ a $\{\tilde a_n\}$ tohoto prostoru (jde tedy o řešení dané rovnice), nechť $u,v\in\mathbb{R}$. Pak příslušná lineární kombinace $u\{a_n\}+v\{\tilde a_n\}$ splňuje pro každé $n \geq n_0$

$$(ua_{n+k} + v\tilde{a}_{n+k}) + \sum_{i=0}^{k-1} c_i(n)(ua_{n+i} + v\tilde{a}_{n+i}) = ua_{n+k} + v\tilde{a}_{n+k} + u\sum_{i=0}^{k-1} c_i(n)a_{n+i} + v\sum_{i=0}^{k-1} c_i(n)\tilde{a}_{n+i}$$
$$= u\Big(a_{n+k} + \sum_{i=0}^{k-1} c_i(n)a_{n+i}\Big) + v\Big(\tilde{a}_{n+k} + \sum_{i=0}^{k-1} c_i(n)\tilde{a}_{n+i}\Big) = u\cdot 0 + v\cdot 0 = 0,$$

takže také řeší danou rovnici a tedy $u\{a_n\} + v\{\tilde{a}_n\} \in M$. $M$ je proto lineární podprostor lineárního prostoru všech posloupností, tudíž lineární prostor.

2) Teď ukážeme, že dimenze $M$ je $k$.

Nechť $\{a_{1,n}\}$ je řešení dané rovnice a počátečních podmínek $a_{n_0} = 1$, $a_{n_0+1} = 0, \dots, a_{n_0+k-1} = 0$.

Nechť $\{a_{2,n}\}$ je řešení dané rovnice a počátečních podmínek $a_{n_0} = 0$, $a_{n_0+1} = 1$, $a_{n_0+2} = 0, \dots, a_{n_0+k-1} = 0$.

Postupujeme obdobně dále a dostaneme $k$ řešení, přičemž $\{a_{i,n}\}$ řeší počáteční podmínky $a_{n_0+i-1} = 1$ a $a_{n_0+j-1} = 0$ pro $j \neq i$, $1 \leq j \leq k$. Tato řešení existují dle Věty 10a.2. Všimněte si, že řešení se doplňují. První z těchto řešení začíná $\{1, 0, 0, 0, \dots\}$, druhé začíná $\{0, 1, 0, 0, \dots\}$, třetí začíná $\{0, 0, 1, 0, 0, \dots\}$, takže kdykoliv se u těchto posloupností podíváme na stejný člen (první, druhý až $k$-tý), tak vždy pouze jedna z nich má hodnotu 1 a ostatní mají hodnotu 0. To bude za chvíli velice důležité.

a) Tvrdíme, že tato řešení jsou lineárně nezávislá. Předpokládejme, že $\sum\limits_{i=1}^{k} u_i\{a_{i,n}\} = \{0\}$, neboli nechť se jistá lineární kombinace těchto řešení rovná nulové posloupnosti, tedy nulovému prvku prostoru $M$. Potřebujeme ukázat, že pak nutně $u_i = 0$ pro každé $i$. Vezměme tedy $j$ z množiny $\{1, 2, \dots, k\}$ a podívejme se v té rovnosti nalevo i napravo na $j$-tý prvek posloupnosti, neboli na prvek posloupnosti s indexem $n_0 + j - 1$. Každý prvek nulové posloupnosti je prostě 0, proto

$$\sum_{i=1}^{k} u_i a_{i,n_0+j-1} = 0$$

Na výraz nalevo aplikujeme počáteční podmínky pro jednotlivá řešení. Jak už jsme diskutovali, jen jediné z těchto řešení má nenulový svůj $j$-tý člen (člen s indexem $n_0 + j - 1$), jde o řešení $\{a_{j,n}\}$ a ten člen je roven $a_{j,n+j-1} = 1$. Proto ta rovnice ve skutečnosti zní $u_j = 0$, přesně jak jsme potřebovali.

Dokázali jsme, že z rovnosti $\sum\limits_{i=1}^{k} u_i\{a_{i,n}\} = \{0\}$ už nutně vyplývá $u_i = 0$ pro všechna $i$, proto je množina $\big\{\{a_{i,n}\}_{n=n_0}^{\infty};\ i = 1, 2, \dots, k\big\}$ lineárně nezávislá. Z toho mimo jiné plyne, že $\dim(M) \geq k$.

b) Nyní dokážeme, že oněch $k$ posloupností také generuje $M$, tedy je to vlastně báze. Vezměme nějaké libovolné řešení $\{a_n\} \in M$, potřebujeme ukázat, že se dá vyjádřit jako lineární kombinace našich $k$ posloupností. Podívejme se na lineární kombinaci $\{\tilde{a}_n\} = \sum\limits_{i=1}^{k} a_{n_0+i-1}\{a_{i,n}\}$ (zde tedy používáme prvních $k$ členů onoho daného řešení jako koeficienty lineární kombinace). Nechť $j$ je nějaké číslo mezi 1 a $k$. Jaký je $j$-tý člen posloupnosti $\{\tilde{a}_n\}$, tj. člen $\tilde{a}_{n_0+j-1}$? Jak jsme již diskutovali, ona řešení zahrnutá v dané lineární kombinaci jsou nenulová vždy pouze jednou, takže dostáváme

$$\tilde{a}_{n_0+j-1} = \sum_{i=1}^{k} a_{n_0+i-1} a_{i,n_0+j-1} = a_{n_0+j-1} \cdot 1 = a_{n_0+j-1}.$$

To znamená, že posloupnosti $\{\tilde{a}_n\}$ a $\{a_n\}$ jsou obě řešením dané rovnice a mají také shodné všechny ze svých prvních $k$-členů, proto se podle Důsledku 10a.3 rovnají, tedy $\{a_n\} = \{\tilde{a}_n\} = \sum\limits_{i=1}^{k} a_{n_0+i-1}\{a_{i,n}\}$.

Ukázali jsme, že libovolné řešení dané rovnice (tj. libovolný prvek množiny $M$) lze vyjádřit jako lineární kombinaci posloupností $\{a_{i,n}\}_{n=n_0}^{\infty}$. Spolu s lineární nezávislostí to znamená, že množina $\{\{a_{i,n}\}_{n=n_0}^{\infty};\ i = 1, 2 \dots, k\}$ je báze prostoru $M$ a proto $\dim(M) = k$.

$\square$

**!** Jaký to má důsledek? Lineární algebra nabízí zajímavou zkratku. Stačí najít nějakou bázi prostoru těchto řešení, jinými slovy, stačí najít nějakých $k$ řešení $\{a_{i,n}\}_{n=n_0}^{\infty}$ této homogenní rovnice tak, aby byly lineárně nezávislé, a už známe všechna řešení, jmenovitě se dají vyjádřit jako lineární kombinace $\sum\limits_{i=1}^{k} u_i\{a_{i,n}\} = \Big\{\sum\limits_{i=1}^{k} u_i a_{i,n}\Big\}_{n=n_0}^{\infty}$. Všimněte si, že se tam objevuje $k$ parametrů $u_i$, což přesně souhlasí, dostáváme tak obecné řešení dané homogenní rovnice.

Na něco podobného jsme narazili u králíků (příklad 10a.b). Rovnice $a_{n+1}-4a_n=0$ je homogenní rovnice 1. řádu, podle této věty je prostor jejích řešení jednorozměrný. My jsme zjistili, že všechny posloupnosti $\{c4^k\}=c\{4^k\}$ jsou řešeními, toto jedno řešení $\{4^k\}$ tedy představuje jednoprvkovou bázi prostoru řešení a vše souhlasí.

To je krásné, ale zatím je to jen samá teorie, pořád ještě nevíme, jak vlastně nějaké to řešení najít. Abychom to napravili, musíme se omezit na ještě pěknější typ rovnic.

## Cvičení

**Cvičení 10a.1** (rutinní)**:** Dokažte, že daná posloupnost řeší danou rovnici, případně i s počátečními podmínkami:
(i) $\{2^n\}_{n=3}^{\infty}$, $a_{n+2}-4a_n=0$ pro všechna $n\geq 3$;
(ii) $\{2^n\}_{n=1}^{\infty}$, $a_{n+2}+a_{n+1}-a_n=5\cdot 2^n$ pro všechna $n\geq 1$, $a_1=2$, $a_2=4$;
(iii) $\{n2^n+1\}_{n=0}^{\infty}$, $a_{n+2}-3a_{n+1}+2a_n=2\cdot 2^n$ pro všechna $n\geq 0$, $a_0=1$, $a_1=3$;
(iv) $\{2^n+3^n\}_{n=0}^{\infty}$, $a_{n+2}-5a_{n+1}+6a_n=0$ pro všechna $n\geq 0$;
(v) $\{2^n+n\}_{n=1}^{\infty}$, $a_{n+2}-a_{n+1}-2a_n=1-2n$ pro všechna $n\geq 1$, $a_1=3$, $a_2=6$.

**Řešení:**
**10a.1:** Stačí dosadit, nejlépe začít levou stranou:
(i): $a_{n+2}-4a_n=2^{n+2}-4\cdot 2^n=4\cdot 2^n-4\cdot 2^n=0$ pro $n\geq 3$;
(ii): $2^{n+2}+2^{n+1}-2^n=4\cdot 2^n+2\cdot 2^n-2^n=5\cdot 2^n$ pro $n\geq 1$, počáteční podmínky: $a_1=2^1=2$, $a_2=2^2=4$ souhlasí;
(iii): $a_{n+2}-3a_{n+1}+2a_n=(2^{n+2}(n+2)+1)-3(2^{n+1}(n+1)+1)+2(2^n n+1)$
$=4n2^n+8\cdot 2^n+1-6n2^n-6\cdot 2^n-3+2n2^n+2=2\cdot 2^n$ pro $n\geq 0$, počáteční podmínky: $a_0=2^0\cdot 0+1=1$, $a_1=2^1\cdot 1+1=3$ souhlasí;
(iv): $a_{n+2}-5a_{n+1}+6a_n=(2^{n+2}+3^{n+2})-5(2^{n+1}+3^{n+1})+6(2^n+3^n)=4\cdot 2^n+9\cdot 3^n-10\cdot 2^n-15\cdot 3^n+6\cdot 2^n+6\cdot 3^n=0$ pro $n\geq 0$;
(v): $a_{n+2}-a_{n+1}-2a_n=(2^{n+2}+(n+2))-(2^{n+1}+(n+1))-2(2^n+n)=4\cdot 2^n+n+2-2\cdot 2^n-n-1-2\cdot 2^n-2n=1-2n$ pro $n\geq 1$, počáteční podmínky: $a_1=2^1+1=3$, $a_2=2^2+2=6$ souhlasí.

## 10b. Rovnice s konstantními koeficienty

Zde se konečně naučíme rovnice řešit. Nejprve se ale musíme vzdát možnosti, že by koeficienty rovnice závisely na $n$.

**!**

**Definice**
**Lineární rekurentní rovnice s konstantními koeficienty** je libovolná rovnice ve tvaru
$$a_{n+k}+c_{k-1}a_{n+k-1}+\dots+c_1a_{n+1}+c_0a_n=b_n \quad \text{pro všechna } n\geq n_0,$$
kde $n_0\in\mathbb{Z}$, $c_i\in\mathbb{R}$ pro $i=0,\dots,k-1$ jsou pevně zvolená čísla a $\{b_n\}_{n=n_0}^{\infty}$ je pevně zvolená posloupnost reálných čísel.

S takovými rovnicemi se už dá něco dělat. Všimněte si, že příklady 10.a, 10.c a 10a.b jsou tohoto typu.

**!**

**Definice**
Nechť je dána lineární rekurentní rovnice s konstantními koeficienty
$$a_{n+k}+c_{k-1}a_{n+k-1}+\dots+c_1a_{n+1}+c_0a_n=b_n \text{ pro všechna } n\geq n_0.$$
Její **charakteristický polynom** (**characteristic polynomial**) je definován jako polynom
$$p(\lambda)=\lambda^k+c_{k-1}\lambda^{k-1}+\dots+c_1\lambda+c_0.$$
Kořeny charakteristického polynomu se nazývají **charakteristická čísla**, popřípadě **vlastní čísla** dané rovnice (**characteristic numbers/roots** or **eigenvalues**).

K získání charakteristických čísel potřebujeme vyřešit rovnici $\lambda^k+c_{k-1}\lambda^{k-1}+\dots+c_1\lambda+c_0=0$, které se také říká **charakteristická rovnice** (**characteristic equation**).

V příkladě s králíky jsme měli rovnici $a_{n+1}-4a_n=0$ a řešení $\{4^n\}$, podle nové definice také máme charakteristický polynom $\lambda-4$ a charakteristické číslo $\lambda=4$. Tato shoda není náhoda.

! **Fakt 10b.1.**
Jestliže je $\lambda$ charakteristickým číslem dané homogenní lineární rekurentní rovnice s konstantními koeficienty
$$a_{n+k} + c_{k-1}a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0 \quad \text{pro všechna } n \geq n_0,$$
pak je geometrická posloupnost $\{\lambda^n\}_{n=n_0}^{\infty}$ jejím řešením.

**Důkaz** (poučný)**:** Dosadíme posloupnost $a_n = \lambda^n$ do dané rovnice (začneme levou stranou). Protože je $\lambda$ nulovým bodem charakteristického polynomu, dostaneme pro libovolné $n \geq n_0$
$$\begin{aligned} a_{n+k} + c_{k-1}a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n &= \lambda^{n+k} + c_{k-1}\lambda^{n+k-1} + \cdots + c_1\lambda^{n+1} + c_0\lambda^n \\ &= \lambda^n(\lambda^k + c_{k-1}\lambda^{k-1} + \cdots + c_1\lambda + c_0) = \lambda^n p(\lambda) = \lambda^n \cdot 0 = 0. \end{aligned}$$
Tato posloupnost tedy řeší danou rovnici.

□

Jak víme, polynom stupně $k$ má $k$ kořenů (pokud bereme i komplexní a včetně násobnosti). Pro nalezení báze řešení rovnice řádu $k$ zase potřebujeme $k$ posloupností. To vypadá slibně, nejprve snadný případ:

! **Věta 10b.2.**
Uvažujme homogenní lineární rekurentní rovnici s konstantními koeficienty řádu $k$
$$a_{n+k} + c_{k-1}a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0 \quad \text{pro všechna } n \geq n_0.$$
Jestliže má $k$ různých charakteristických čísel $\lambda_i$, pak posloupnosti $\{\lambda_i^n\}_{n=n_0}^{\infty}$ tvoří bázi prostoru řešení dané rovnice.

To je velice příjemné tvrzení, protože obecné řešení této rovnice pak je $\sum\limits_{i=1}^{k} u_i\{\lambda_i^n\} = \Big\{\sum\limits_{i=1}^{k} u_i\lambda_i^n\Big\}$.

**Důkaz** (drsný)**:** Dané posloupnosti tvoří řešení dle předchozího tvrzení, stačí tedy dokázat, že jsou nezávislé, a bude to i báze, neboť je jich tolik, kolik je dimenze prostoru řešení (Věta 10a.5).

Potřebujeme ukázat, že rovnice $\sum\limits_{i=1}^{k} u_i\{\lambda_i^n\} = \{0\}$ nutně vede na $u_i = 0$ pro všechna $i$. Jako obvykle se stačí podívat na prvních $k$ členů zúčastněných posloupností, dostaneme následující soustavu rovnic:
$$\begin{array}{llllll} u_1\lambda_1^{n_0} & +u_2\lambda_2^{n_0} & +\cdots+u_k\lambda_k^{n_0} & =0 \\ u_1\lambda_1^{n_0+1} & +u_2\lambda_2^{n_0+1} & +\cdots+u_k\lambda_k^{n_0+1} & =0 \\ u_1\lambda_1^{n_0+2} & +u_2\lambda_2^{n_0+2} & +\cdots+u_k\lambda_k^{n_0+2} & =0 \\ \vdots & \vdots & \vdots & \vdots \\ u_1\lambda_1^{n_0+k-1} & +u_2\lambda_2^{n_0+k-1} & +\cdots+u_k\lambda_k^{n_0+k-1} & =0 \end{array} \Longrightarrow \begin{array}{llll} u_1 & +u_2 & +\cdots+u_k & =0 \\ u_1\lambda_1 & +u_2\lambda_2 & +\cdots+u_k\lambda_k & =0 \\ u_1\lambda_1^2 & +u_2\lambda_2^2 & +\cdots+u_k\lambda_k^2 & =0 \\ \vdots & \vdots & \vdots & \vdots \\ u_1\lambda_1^{k-1} & +u_2\lambda_2^{k-1} & +\cdots+u_k\lambda_k^{k-1} & =0 \end{array}$$
Potřebujeme ukázat, že jejím jediným řešením je to triviální, což znamená, že potřebujeme ukázat, že matice soustavy
$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_k \\ \lambda_1^2 & \lambda_2^2 & \cdots & \lambda_k^2 \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_1^{k-1} & \lambda_2^{k-1} & \cdots & \lambda_k^{k-1} \end{pmatrix}$$
je regulární. To je ale pro navzájem různá $\lambda_i$ standardní fakt z lineární algebry, jde o tzv. Vandermontův determinant.

□

! **Příklad 10b.a:** V příkladě 10.c jsme se dívali na funkci zadanou induktivně podmínkami $f(1) = 3$, $f(2) = -1$ a $f(n) = f(n-1) + 6f(n-2)$ pro $n \geq 3$. V řeči posloupností to je $f_n = f_{n-1} + 6f_{n-2}$, do správného tvaru lineární rovnice to převedeme například substitucí $m = n-2$ neboli $n = m+2$. Dostáváme homogenní lineární rovnici $f_{m+2} - 3a_{m+1} + 2f_m = 0$ pro $m \geq 1$, hledáme řešení splňující počáteční podmínky $f_1 = 3$, $f_2 = -1$. Protože je rovnice 2. řádu, počet podmínek souhlasí.

Řešení: Nejprve najdeme obecné řešení. Určíme charakteristický polynom:
$$p(\lambda) = \lambda^2 - \lambda - 6 = (\lambda-3)(\lambda+2).$$
Z rovnice $(\lambda-3)(\lambda+2) = 0$ dostáváme charakteristická čísla $\lambda = 3, -2$ a obecné řešení $\{u \cdot 3^n + v \cdot (-2)^n\}_{n=1}^{\infty}$, tedy $f_n = 3^n u + (-2)^n v$ pro libovolná $u, v \in \mathbb{R}$.

Teď mezi těmito nekonečně mnoha řešeními najdeme to, které splňuje počáteční podmínky. Ty vyžadují následující:

$$\begin{aligned} f_1 &= 3^1u + (-2)^1v = 3 \\ f_2 &= 3^2u + (-2)^2v = -1 \end{aligned} \quad\Longrightarrow\quad \begin{aligned} 3u - 2v &= 3 \\ 9u + 4v &= -1. \end{aligned}$$

Odtud hravě vykutáme, že $u = \frac{1}{3}$, $v = -1$. Řešení dané úlohy tedy je $\left\{\frac{1}{3}\cdot 3^n + (-1)\cdot(-2)^n\right\}_{n=1}^{\infty}$ neboli (přejdeme k funkcím dle zadání) $f(n) = 3^{n-1} - (-2)^n$ pro $n \geq 1$.

Po vyřešení úlohy bývá dobré udělat zkoušku, tj. ověřit, že to, co jsme našli, je opravdu řešení.
Takže: $f(1) = 3^0 - (-2) = 3$ a $f(2) = 3^1 - (-2)^2 = -1$, to souhlasí, teď se podíváme na induktivní rovnici:

$$\begin{aligned} f(n) + 6f(n-1) &= [3^{n-1} - (-2)^n] + 6[3^{n-2} - (-2)^{n-1}] = 3^{n-1} - (-2)^n + 2\cdot 3^{n-1} + 3\cdot(-2)^n \\ &= 3\cdot 3^{n-1} + 2\cdot(-2)^n = 3^{(n+1)-1} - (-2)^{n+1} = f(n+1). \end{aligned}$$

Poznámka: Pokud bychom při zápisu obecného řešení chtěli použít metody z lineární algebry, mohli bychom množinu všech řešení zapsat například těmito způsoby:

$$\big\{\{3^{n-1}u + (-2)^n v\}_{n=1}^{\infty};\ u,v\in\mathbb{R}\big\} = \big\{u\{3^{n-1}\}_{n=1}^{\infty} + v\{(-2)^n\}_{n=1}^{\infty};\ u,v\in\mathbb{R}\big\} = \big\langle\{3^{n-1}\}_{n=1}^{\infty}, \{(-2)^n\}_{n=1}^{\infty}\big\rangle.$$

△

**!** Nejčastěji řešíme dva druhy úloh.

1) Chceme obecné řešení dané rovnice (tedy vlastně chceme znát množinu všech řešení). Pak najdeme bázi prostoru řešení pomocí charakteristických čísel a obecné řešení dostaneme jako lineární kombinaci této báze.

2) Chceme řešení rovnice, které navíc splňuje dané počáteční podmínky. Jak jsme právě viděli, tento druhý typ úlohy se obvykle řeší ve dvou krocích. Nejprve se najde obecné řešení podle postupu 1) a pak se určí hodnoty konstant tak, aby výsledné řešení splňovalo počáteční podmínky. Tento druhý krok je v zásadě triviální, prostě si napíšeme, co od obecného řešení chceme (počáteční podmínky), pak vyřešíme vzniklých $k$ rovnic o $k$ neznámých a je to.

**Příklad 10b.b:** Zde zkusíme jiný pohled na králíky. Na začátku prvního měsíce dostaneme párek čerstvě narozených králíků. Nechť $F_n$ je počet párů na začátku $n$-tého měsíce od obdržení. Podle jakých pravidel se králíci množí? Uvažujme tyto zásady:

- králíci se začnou množin ve chvíli, kdy jsou jim 2 měsíce;
- když se začnou množit, tak pak mají každý měsíc jeden pár mladých;
- králíci nikdy neumřou (jsou to matematičtí králíci).

Jak to pak s jejich počty vypadá? Na začátku 1. měsíce je $F_1 = 1$ pár. Pořád je ještě mladý, takže i na začátku druhého měsíce je jen $F_2 = 1$ pár. Pak se ale začne množit a na začátku dalšího měsíce už má mladé, proto $F_3 = 1 + 1 = 2$. Na začátku 4. měsíce se ten první pár znovu zmnožil, ale druhý je ještě mladý, takže $F_4 = 2 + 1 = 3$. A tak dále, jak to vlastně funguje?

Chceme-li znát, kolik bude králíků na začátku měsíce $n+1$, tak výchozím stavem je samozřejmě stav předchozí, tedy $F_n$. K tomu se musí přičíst nové přírůstky, což se přesně rovná počtu párů, kterým je na začátku měsíce $n+1$ alespoň dva měsíce, neboli počtu párů, které už tu byly na začátku měsíce $n-1$. Dostáváme $F_{n+1} = F_n + F_{n-1}$, což je rovnice udávající Fibonacciho posloupnost, viz příklad 9a.c. Přesně takovouto úvahou (včetně nesmrtelnosti králíků) k ní ctihodný Leonardo z Pisy známý též jako Fibonacci přišel.

Přepíšeme si to jako $F_{n+2} - F_{n+1} - F_n = 0$ a tuto homogenní lineární rekurentní rovnici 2. řádu s počátečními podmínkami $F_1 = 1$, $F_2 = 1$ hravě vyřešíme.

Nejprve obecné řešení: Z $p(\lambda) = \lambda^2 - \lambda - 1 = 0$ dostáváme charakteristická čísla $\lambda = \dfrac{1\pm\sqrt{5}}{2}$. Obecné řešení dané rovnice je tedy $F_n = u\Big(\dfrac{1+\sqrt{5}}{2}\Big)^n + v\Big(\dfrac{1-\sqrt{5}}{2}\Big)^n$.

Aplikujeme počáteční podmínky:

$$\begin{aligned} F_1 &= u\frac{1+\sqrt{5}}{2} + v\frac{1-\sqrt{5}}{2} = 1 \\ F_2 &= u\Big(\frac{1+\sqrt{5}}{2}\Big)^2 + v\Big(\frac{1-\sqrt{5}}{2}\Big)^2 = 1 \end{aligned} \quad\Longrightarrow\quad \begin{aligned} u(1+\sqrt{5}\,) + v(1-\sqrt{5}\,) &= 2 \\ u(3+\sqrt{5}\,) + v(3-\sqrt{5}\,) &= 2 \end{aligned}$$

Když rovnice odečteme, dostaneme $u + v = 0$, odtud $v = -u$, dosadíme do první rovnice a dostaneme $2u\sqrt{5} = 2$, tedy $u = \dfrac{1}{\sqrt{5}} = -v$.

Závěr: Fibonacciho posloupnost je dána explicitním vzorcem $F_n = \dfrac{1}{\sqrt{5}}\Big(\dfrac{1+\sqrt{5}}{2}\Big)^n - \dfrac{1}{\sqrt{5}}\Big(\dfrac{1-\sqrt{5}}{2}\Big)^n$.

Tato posloupnost možná není nejlepším modelem pro množení králíků, ale objevuje se při zkoumání jevů překvapivě často. Již jsme ji viděli při odhadu rychlosti Euklidova algoritmu (Věta 6a.18) a ještě ji potkáme jak v této kapitole, tak v příkladě 11b.f.

Zajímavé je číslo $\varphi = \frac{1+\sqrt{5}}{2} = 1.618...$ Je to přesně hodnota zlatého řezu, další úžasná náhoda. Protože je to to větší z charakteristických čísel, je dominantní a tedy v jazyce kapitoly 9b můžeme říct, že $F_n = \Theta(\varphi^n)$.

Možná vám vrtá hlavou, co by se stalo, kdybychom do toho zapracovali smrtelnost králíků. Zkusme třeba toto:
- na konci čtvrtého měsíce králíci umírají.

Jaký model z toho vyjde teď? To je trochu komplikovanější, je to možná lépe vidět, když si zavedeme posloupnost $b_n$, která říká, kolik párů králíků se narodilo na začátku měsíce $n$. Toto číslo se rovná počtu párů, kterým je v té chvíli dva či tři měsíce, protože ti mladší ještě nemohou a ti starší už mají přesně opačné starosti. Dostáváme vztah $b_n = b_{n-2} + b_{n-3}$.

Aktuální stav na začátku měsíce $n$, označme jej $c_n$, je dán součtem počtu králíků, kteří se právě narodili či jsou měsíc, dva a tři měsíce staří, tedy $c_n = b_n + b_{n-1} + b_{n-2} + b_{n-3}$. Když na všechny $b_i$ aplikujeme již odvozený vztah, dostaneme

$$\begin{aligned} c_n &= (b_{n-2} + b_{n-3}) + (b_{n-3} + b_{n-4}) + (b_{n-4} + b_{n-5}) + (b_{n-5} + b_{n-6}) \\ &= (b_{n-2} + b_{n-3} + b_{n-4} + b_{n-5}) + (b_{n-3} + b_{n-4} + b_{n-5} + b_{n-6}) = c_{n-2} + c_{n-3}. \end{aligned}$$

Posun indexu dává $c_{n+1} = c_{n-1} + c_{n-2}$, je to tedy podobné Fibonacciho vztahu, počátek posloupnosti dokonce s Fibonacciho posloupností souhlasí, $c_1 = c_2 = 1$, $c_3 = 2$, protože v té době se ještě smrtelnost neprojevila. Dostáváme posloupnost $\{1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, \dots\}$, která evidentně (a logicky) roste pomaleji než ta pro Fibonacciho nesmrtelné králíky. Jak rychle vlastně roste?

Přepis dává homogenní lineární rekurentní rovnici $c_{n+3} - c_{n+1} - c_n = 0$ třetího řádu. Bohužel, její charakteristická rovnice $\lambda^3 - \lambda - 1 = 0$ nemá žádné „pěkné“ kořeny. To ale nevadí. K určení rychlosti růstu nám vlastně stačí znát kořen, který je v absolutní hodnotě největší, ten se dá určit i přibližně některou z oblíbených metod (Newtonova, bisekce), vychází 1.325... To znamená, že $c_n$ je přibližně $\Theta((1.325)^n)$.

△

Zatím jsme měli v příkladech štěstí a vždy jsme dostali $k$ různých charakteristických čísel. Na to se ale nedá spoléhat a v případě opakovaného kořene zatím nevíme, co dělat. Navíc ani různé kořeny ještě neznamenají úspěch, protože ony mohou být komplexní, zatímco my zde řešíme rovnice v reálném oboru (konec konců uvažujeme jen reálné koeficienty rovnic). S těmito dvěma problémy se musíme naučit vyrovnat, začneme tím prvním.

**Fakt 10b.3.**
Nechť je dána homogenní lineární rekurentní rovnice s konstantními koeficienty. Jestliže je $\lambda$ její charakteistické číslo a má násobnost $m$ jako kořen charakteristického polynomu, pak posloupnosti $\{\lambda^n\}, \{n\lambda^n\}, \dots, \{n^{m-1}\lambda^n\}$ jsou řešením dané rovnice a tvoří lineárně nezávislou množinu.

Pro zkrácení budeme v takové situaci prostě říkat, že $\lambda$ je charakteristické číslo násobnosti $m$.

**Důkaz** (náznak, poučný)**:** Důkaz tohoto faktu je obecně drobet dobrodružnější, proto ukážeme, proč to platí pro dvojnásobný kořen. Použijeme obecný postup, který funguje i pro charakteristická čísla vyšší násobnosti.

Mějme tedy rovnici $a_{n+k} + \sum\limits_{i=0}^{k-1} c_i a_{n+i} = 0$, kde $c_0 \neq 0$. Její charakteristický polynom je $p(\lambda) = \lambda^k + \sum\limits_{i=0}^{k-1} c_i \lambda^i$. Jestliže je $\lambda_0$ alespoň dvojnásobný kořen, pak teorie polynomů (nebo obecně funkcí) říká, že platí nejen $p(\lambda_0) = 0$, ale také $p'(\lambda_0) = 0$ (zde $p'$ je derivace). To budeme záhy potřebovat.

1) Potřebujeme dokázat, že posloupnosti $\{\lambda_0^n\}$ a $\{n\lambda_0^n\}$ řeší danou rovnici. Pro první posloupnost už to víme, zbývá tedy ověřit, že i ta druhá funguje. Dosadíme do levé strany rovnice.

$$\begin{aligned} a_{n+k} + \sum_{i=0}^{k-1} c_i a_{n+i} &= (n+k)\lambda_0^{n+k} + \sum_{i=0}^{k-1} c_i (n+i)\lambda_0^{n+i} = n\lambda_0^{n+k} + k\lambda_0^{n+k} + \sum_{i=0}^{k-1} c_i n \lambda_0^{n+i} + \sum_{i=0}^{k-1} c_i i \lambda_0^{n+i} \\ &= n\lambda_0^{n+k} + \sum_{i=0}^{k-1} c_i n \lambda_0^{n+i} + k\lambda_0^{n+k} + \sum_{i=0}^{k-1} c_i i \lambda_0^{n+i} \\ &= n\lambda_0^n \Big( \lambda_0^k + \sum_{i=0}^{k-1} c_i \lambda_0^i \Big) + \lambda_0^{n+1} \Big( k\lambda_0^{k-1} + \sum_{i=0}^{k-1} c_i i \lambda_0^{i-1} \Big) \\ &= n\lambda_0^n p(\lambda_0) + \lambda_0^{n+1} p'(\lambda_0) = n\lambda_0^n \cdot 0 + \lambda_0^{n+1} \cdot 0 = 0. \end{aligned}$$

Posloupnost $\{n\lambda_0^n\}$ tedy splňuje danou rovnici.

2) Teď ještě potřebujeme dokázat lineární nezávislost těch dvou posloupností. Podobně jako u předchozího důkazu nezávislosti se to nakonec převede na otázku, zda je matice $\begin{pmatrix} 1 & 0 \\ \lambda & \lambda \end{pmatrix}$ regulární. Protože $c_0 \neq 0$, musí být i $\lambda_0 \neq 0$ a matice regulární je.

3) Obecně by se muselo dokázat, že $\{n^{m-1}\lambda^n\}$ je řešením, pokud má $\lambda$ násobnost alespoň $m$, což by se dělalo podobně, jen to dá víc práce, použilo by se přitom faktu, že pro kořen $\lambda_0$ násobnosti $m$ se v bodě $\lambda_0$ vynulují i derivace $p'$, $p''$, … ,$p^{(m-1)}$.

Nezávislost by se pak redukovala na otázku regulárnosti matice

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ \lambda & \lambda & \lambda & \dots & \lambda \\ \lambda^2 & 2\lambda^2 & 2^3\lambda^2 & \dots & 2^{m-1}\lambda^2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \lambda^{m-1} & (m-1)\lambda^{m-1} & (m-1)^3\lambda^{m-1} & \dots & (m-1)^{m-1}\lambda^{m-1} \end{pmatrix}$$

I to je něco, na co nám kladně odpoví lineární algebra.

□

Vidíme tedy, že pokud má některý kořen charakteristického polynomu vyšší násobnost $m$, tak sice bude o $m-1$ méně různých kořenů ve Větě 10b.2, ale stejný počet nezávislých řešení zase přibyde díky tomu násobení členem $n^i$. Celkový počet řešení je tedy pořád správný, ale ještě zbývá dokázat, že tato nová řešení typu $\{n^i\lambda^n\}$ nezkazí nezávislost po přidání k posloupnostem pocházejícím od ostatních $\lambda$.

**!** **Věta 10b.4.**
Nechť je dána homogenní lineární rekurentní rovnice s konstantními koeficienty řádu $k$. Nechť jsou $\lambda_1, \dots, \lambda_M$ její různá charakteristická čísla, přičemž každé $\lambda_i$ má násobnost $m_i \in \mathbb{N}$. Pak je množina

$$\big\{\{\lambda_1^n\}, \{n\lambda_1^n\}, \dots, \{n^{m_1-1}\lambda_1^n\}, \{\lambda_2^n\}, \{n\lambda_2^n\}, \dots, \{n^{m_2-1}\lambda_2^n\}, \dots, \{\lambda_M^n\}, \{n\lambda_M^n\}, \dots, \{n^{m_M-1}\lambda_M^n\}\big\}$$

bází prostoru řešení dané rovnice.

Vlastně už toho hodně máme dokázaného, zbývá lineární nezávislost výsledné množiny řešení, která se zase redukuje na regularitu jisté matice a to je práce pro lineární algebru (tentokráte opravdu drsná). Stejně jako autoři většiny učebnic na toto téma i my toho čtenáře (a sebe) ušetříme. Opravdu zvědavého čtenáře odkážeme na nějakou podrobnější učebnici diferenciálních rovnic, kde se u homogenních lineárních rovnic dělá prakticky totéž.

**!** **Příklad 10b.c:** Najdeme obecné řešení rovnice $a_{n+3} - a_{n+2} - a_{n+1} + a_n = 0$ pro všechna $n \geq -2$.

Řešení: Je to homogenní lineární rekurentní rovnice s konstantními koeficienty, Věta 10b.4 tedy dává návod k řešení.

Charakteristický polynom je $p(\lambda) = \lambda^3 - \lambda^2 - \lambda + 1$. Kořeny polynomu třetího stupně se sice dají získat pomocí vzorců, ale ty jsou díky své komplikovanosti vysoce nepopulární. Toto je školní příklad, nenašel by se nějaký pěkný kořen? Zkusíme dosadit malá celá čísla, 0 nefunguje, ale 1 ano. Máme první kořen a teď odloupneme z $p$ příslušný kořenový faktor:

$$p(\lambda) = (\lambda - 1)(\lambda^2 - 1).$$

A je to jasné,

$$p(\lambda) = (\lambda - 1)(\lambda - 1)(\lambda + 1) = (\lambda - 1)^2(\lambda + 1).$$

Máme tedy charakteristická čísla $\lambda = 1$ (dvojnásobné) a $\lambda = -1$ (jednoduché). Podle Věty 10b.4 proto bude množina

$$\big\{\{1^n\}_{n=-2}^{\infty}, \{n1^n\}_{n=-2}^{\infty}, \{(-1)^n\}_{n=-2}^{\infty}\big\}$$

bází prostoru všech řešení, dostáváme obecné řešení $\{u \cdot 1^n + v \cdot n1^n + w \cdot (-1)^n\}_{n=-2}^{\infty} = \{u + vn + w(-1)^n\}_{n=-2}^{\infty}$ pro $u, v, w \in \mathbb{R}$.

Pro úplnost uděláme zkoušku, dosadíme toto řešení do levé strany dané rovnice. Pro $n \geq -2$ máme

$$\begin{aligned} a_{n+3} - a_{n+2} - a_{n+1} + a_n &= (u + v(n+3) + w(-1)^{n+3}) - (u + v(n+2) + w(-1)^{n+2}) \\ &\quad - (u + v(n+1) + w(-1)^{n+1}) + (u + vn + w(-1)^n) \\ &= u + vn + 3v + w(-1)^3(-1)^n - u - vn - 2v - w(-1)^2(-1)^n \\ &\quad - u - vn - v - w(-1)^1(-1)^n + u + vn + w(-1)^n = 0. \end{aligned}$$

△

Teď už tedy teoreticky umíme vyřešit všechny homogenní lineární rovnice s konstantními koeficienty, pokud jsme spokojeni s případnými komplexními posloupnostmi. Jenže my spokojeni nejsme, když zde mluvíme o rovnicích s reálnými koeficienty, tak také očekáváme reálná řešení. Potřebujeme tedy ještě jeden trik.

Je známo, že jestliže je $\lambda = \alpha + \beta i$ kořenem polynomu $p$, pak je jeho kořenem i $\lambda^* = \alpha - \beta i$. V bázi se tedy objeví i dvojice $\{(\alpha+\beta i)^n\}, \{(\alpha-\beta i)^n\}$. Klíčem je podívat se na jejich lineární kombinace $u(\alpha+\beta i)^n + v(\alpha-\beta i)^n$, ukáže se totiž, že mezi nimi je i dostatečný počet reálných posloupností na to, aby nám daly dvojrozměrný lineární prostor (nad reálnými čísly), což je přesně to, co potřebujeme.

Dělá se to takto: Napíšeme $\alpha \pm \beta i = r[\cos(\varphi) \pm i\sin(\varphi)]$, pak $(\alpha \pm \beta i)^n = r^n[\cos(n\varphi) \pm i\sin(n\varphi)]$, a proto

$$(\alpha+\beta i)^n + (\alpha-\beta i)^n = 2r^n\cos(n\varphi),$$
$$(\alpha+\beta i)^n - (\alpha-\beta i)^n = 2ir^n\sin(n\varphi).$$

Vidíme tedy, že pokud vezmeme lineární kombinaci $\frac{1}{2}\{\lambda^n + (\lambda^*)^n\}$, tak dostaneme řešení dané rovnice ve tvaru $\{r^n\cos(n\varphi)\}$, což už je reálná posloupnost, zatímco lineární kombinace $\frac{1}{2i}\{\lambda^n - (\lambda^*)^n\}$ dává další reálné řešení $\{r^n\sin(n\varphi)\}$. Dá se také ukázat, že posloupnosti $\{r^n\cos(n\varphi)\}$ a $\{r^n\sin(n\varphi)\}$ jsou nezávislé. Jaké je tedy ponaučení?

**!** Komplexní kořeny vždy chodí po dvou, takže pokud dostaneme charakteristické číslo $\lambda = r[\cos(\varphi) + i\sin(\varphi)]$, které není reálné, tak použijeme dvojici řešení $\{r^n\cos(n\varphi)\}$ a $\{r^n\sin(n\varphi)\}$. Je-li to kořen vícenásobný, tak obvyklým způsobem dále přihodíme $\{nr^n\cos(n\varphi)\}$ a $\{nr^n\sin(n\varphi)\}$, případně $\{n^2r^n\cos(n\varphi)\}$ a $\{n^2r^n\sin(n\varphi)\}$ až po $\{n^{m-1}r^n\cos(n\varphi)\}$ a $\{n^{m-1}r^n\sin(n\varphi)\}$, kde $m$ je násobnost kořene $\lambda$. Tato řešení (kterých je $2m$) zároveň zastupují řešení pro sdružený kořen $\lambda^*$.

Teď už tedy opravdu umíme vyřešit všechny homogenní lineární rekurentní rovnice s konstantními koeficienty. Mohli bychom to zase potvrdit větou, ale stejně bychom ji nedokazovali, tak raději rovnou zformulujeme algoritmus.

**S Algoritmus 10b.5.** pro řešení homogenní lineární rekurentní rovnice $a_{n+k} + \sum\limits_{i=0}^{k-1} c_i a_{n+i} = 0$, $n \geq n_0$ řádu $k$.

**1.** Sestavte charakteristický polynom $p(\lambda) = \lambda^k + \sum\limits_{i=0}^{k-1} c_i\lambda^i$.

Řešením rovnice $p(\lambda) = 0$ najděte všechna charakteristická čísla dané rovnice.

**2.** Sestavte množinu posloupností $B$ takto:

• pro každé reálné charakteristické číslo $\lambda$ přidejte do $B$ posloupnost $\{\lambda^n\}_{n=n_0}^{\infty}$;

• pro každé reálné charakteristické číslo $\lambda$, jehož násobnost je $m > 1$, přidejte do $B$ rovněž posloupnosti $\{n\lambda^n\}_{n=n_0}^{\infty}, \dots, \{n^{m-1}\lambda^n\}_{n=n_0}^{\infty}$;

• pro každé komplexní charakteristické číslo $\lambda = r[\cos(\varphi) + i\sin(\varphi)]$, které není reálné, přidejte do $B$ posloupnosti $\{r^n\cos(n\varphi)\}_{n=n_0}^{\infty}$ a $\{r^n\sin(n\varphi)\}_{n=n_0}^{\infty}$; pro jeho komplexně sdružené číslo $\lambda^*$ již do $B$ nic nepřidáváme;

• pro každé komplexní charakteristické číslo $\lambda = r[\cos(\varphi) + i\sin(\varphi)]$, které není reálné a jehož násobnost je $m > 1$, přidejte do $B$ posloupnosti $\{nr^n\cos(n\varphi)\}_{n=n_0}^{\infty}, \dots, \{n^{m-1}r^n\cos(n\varphi)\}_{n=n_0}^{\infty}$ a $\{nr^n\sin(n\varphi)\}_{n=n_0}^{\infty}, \dots, \{n^{m-1}r^n\sin(n\varphi)\}_{n=n_0}^{\infty}$; pro jeho komplexně sdružené číslo $\lambda^*$ již do $B$ nic nepřidáváme.

Množina $B$ je bází prostoru řešení.

**3.** Označíme-li $B = \big\{\{a_{1,n}\}, \dots, \{a_{k,n}\}\big\}$, pak je obecné řešení dané rovnice určeno vzorcem $\Big\{\sum\limits_{i=1}^{k} u_i a_{i,n}\Big\}_{n=n_0}^{\infty}$ pro $u_1, \dots, u_k \in \mathbb{R}$.

**4.** Jsou-li dány počáteční podmínky, pak do nich za příslušná $a_j$ pro $j = n_0, \dots, n_0 + k - 1$ dosadíme vzorce $a_j = \sum\limits_{i=1}^{k} u_i a_{i,j}$ a vyřešíme vzniklých $k$ rovnic pro $k$ neznámých $u_i$. Ty po dosazení do obecného řešení určí příslušné partikulární řešení.

△

Hned si to ukážeme na příkladě.

**!** **Příklad 10b.d:** Najdeme řešení rovnice $a_{n+3} - a_{n+2} + a_{n+1} - a_n = 0$ pro všechna $n \geq 13$ s počátečními podmínkami $a_{13} = 1$, $a_{14} = 0$, $a_{15} = 1$.

Řešení: Je to homogenní lineární rekurentní rovnice 3. řádu s konstantními koeficienty a máme tři počáteční podmínky, je to tedy korektně zadaná úloha a použijeme algoritmus 10b.5.

1) Nejprve najdeme obecné řešení. Charakteristický polynom je $p(\lambda) = \lambda^3 - \lambda^2 + \lambda - 1$. Zkusíme nějaký kořen uhádnout, trefíme se s 1. Máme první kořen a rozložíme $p$ příslušným způsobem:

$$p(\lambda) = (\lambda - 1)(\lambda^2 + 1).$$

A je to jasné,

$$p(\lambda) = (\lambda - 1)(\lambda - i)(\lambda + i).$$

Máme tedy jednoduchá charakteristická čísla $\lambda = 1$ a $\lambda = \pm i = 1 \cdot \big[\cos\big(\frac{\pi}{2}\big) \pm i \sin\big(\frac{\pi}{2}\big)\big]$. Podle Algoritmu 10b.5 proto bude množina

$$\Big\{\{1^n\}_{n=13}^{\infty}, \big\{1^n \cos\big(n\tfrac{\pi}{2}\big)\big\}_{n=13}^{\infty}, \big\{1^n \sin\big(n\tfrac{\pi}{2}\big)\big\}_{n=13}^{\infty}\Big\}$$

bází prostoru všech řešení, dostáváme obecné řešení

$$\big\{u \cdot 1^n + v \cdot 1^n \cos\big(n\tfrac{\pi}{2}\big) + w \cdot 1^n \sin\big(n\tfrac{\pi}{2}\big)\big\}_{n=13}^{\infty} = \big\{u + v\cos\big(n\tfrac{\pi}{2}\big) + w\sin\big(n\tfrac{\pi}{2}\big)\big\}_{n=13}^{\infty} \text{ pro } u, v, w \in \mathbb{R}.$$

2) Teď je třeba najít řešení vyhovující počátečním podmínkám. Dosadíme do nich obecné řešení:

$$\begin{aligned} a_{13} &= u + v\cos\big(13\tfrac{\pi}{2}\big) + w\sin\big(13\tfrac{\pi}{2}\big) = 1 & & u + w = 1 \\ a_{14} &= u + v\cos\big(14\tfrac{\pi}{2}\big) + w\sin\big(14\tfrac{\pi}{2}\big) = 0 & \Longrightarrow \quad & u - v = 0 \\ a_{15} &= u + v\cos\big(15\tfrac{\pi}{2}\big) + w\sin\big(15\tfrac{\pi}{2}\big) = 1 & & u - w = 1. \end{aligned}$$

Odtud $u = v = 1$ a $w = 0$. Dosazením do obecného řešení dostáváme hledané partikulární řešení

$$\big\{1 + \cos\big(n\tfrac{\pi}{2}\big)\big\}_{n=13}^{\infty}.$$

Vypadá trochu komplikovaně, ve skutečnosti je to periodická posloupnost $\{1, 0, 1, 2, 1, 0, 1, 2, 1, 0, 1, 2, 1, \dots\}$.

△

Krásný algoritmus. Bohužel funguje jen v případě, že máme snadné rovnice. Jakmile má rovnice řád vyšší než dva, je malá šance, že získáme kořeny charakteristické rovnice. Pak nezbývá než hledat kořeny pomocí numerických metod, které ale dávají hodnoty jen přibližné, nastávají problémy s násobností a vůbec je to nepříjemné. Proto bývá jednodušší hledat numericky přímo řešení.

**10b.6 Nehomogenní rovnice.**

Tak jsme se naučili, jak najít všechna řešení homogenní lineární rovnice. Víme už také, že v případě, že je rovnice nehomogenní, nám stačí nějak najít jedno její řešení, a už zase budeme umět najít všechny. Bohužel, k nalezení toho jednoho řešení obecný algoritmus neexistuje. Zde je třeba spoléhat na náhodu, například existuje jistý typ pravé strany $b_n$, pro který již řešení umíme uhodnout.

**Definice**
Řekneme, že posloupnost $\{b_n\}_{n=n_0}^{\infty}$ je **kvazipolynom** (**quasipolynomial**), jestliže existuje $\lambda \in \mathbb{R}$ a polynom $P(n)$ takový, že $b_n = P(n)\lambda^n$ pro všechna $n \ge n_0$.

Naštěstí pro nás vede mnoho příkladů právě na kvazipolynomiální pravou stranu a můžeme použít následující tvrzení.

**!** **Věta 10b.7.** (řešení pro kvazipolynomiální pravou stranu)
Uvažujme rovnici

$$a_{n+k} + c_{k-1}a_{n+k-1} + \dots + c_1 a_{n+1} + c_0 a_n = b_n \qquad \text{pro všechna } n \ge n_0.$$

Předpokládejme, že existují $\lambda \in \mathbb{R}$ a polynom $P$ takový, že $b_n = P(n)\lambda^n$ pro všechna $n \ge n_0$. Nechť $m$ je násobnost tohoto čísla $\lambda$ jako charakteristického čísla přidružené homogenní rovnice, přičemž $m = 0$ v případě, že toto $\lambda$ vůbec charakteristickým číslem není.
Pak existuje polynom $Q(n)$ stupně stejného jako $P$ takový, že $\{n^m Q(n)\lambda^n\}$ je řešením dané rovnice.

Důkaz je poněkud komplikovanější a vynecháme jej.

Všimněte si, že kvazipolynomy zahrnují i polynomy, volba $\lambda = 1$ totiž dává $b_n = P(n)$. Toto pozorování se ještě bude hodit.

Na první pohled to vypadá jako další teoretické tvrzení o existenci řešení, které při praktickém hledání moc nepomůže, ale není tomu tak. Je pravda, že nám věta neřekla, který polynom $Q$ bude fungovat, ale množina nabízených možností je natolik malá, že si z ní již dokážeme vybrat to správné $Q(n)$.

Slouží k tomu tzv. **metoda odhadu** či **metoda neurčitých koeficientů** (**method of undetermined coefficients**). Uhodneme řešení jako $\{n^m Q(n)\lambda^n\}$, kde jsme za $Q$ napsali obecný polynom vhodného stupně, jeho koeficienty tedy budou sloužit jako parametry. Zbývá najít hodnoty těchto parametrů tak, aby vzniklá posloupnost byla řešením, a to se dělá dosazením posloupnosti do dané rovnice. Tím zjistíme, který polynom funguje, a jsme hotovi. Nejlépe to vysvětlí příklad.

**!** **Příklad 10b.e:** Najdeme obecné řešení rovnice $a_{n+2} - 2a_{n+1} - 3a_n = -9n2^n$ pro $n \ge 0$.
Využijeme Větu 10a.4 a začneme tím snažším.

1) Najdeme obecné řešení rovnice $a_{n+2}-2a_{n+1}-3a_n=0$ pro $n\geq 0$. Charakteristická rovnice $\lambda^2-2\lambda-3=0$ dává charakteristická čísla $-1,3$ a obecné řešení $\{u\cdot(-1)^n+v\cdot 3^n\}_{n=0}^{\infty}$.

2) Teď musíme najít nějaké řešení dané rovnice. Máme štěstí, pravá strana je kvazipolynom, kde $\lambda=2$ a $P(n)=-9n$ je polynom stupně 1. Protože číslo $\lambda=2$ není charakteristickým číslem přidružené homogenní rovnice (viz první část), bude $m=0$ a člen $n^0=1$ vlastně z řešení vypadne, to je také příjemné. Vidíme tedy, že určitě existuje nějaké řešení ve tvaru $\{n^0\cdot Q(n)\cdot 2^n\}=\{Q(n)2^n\}$, kde $Q$ je jistý polynom stupně 1. Takový obecný polynom je dán jako $Q(n)=An+B$, takže Věta 10b.7 vlastně garantuje, že existují nějaké konstanty $A,B$ takové, že $\{(An+B)2^n\}$ je řešením dané rovnice. Tyto neznámé konstanty najdeme tak, že dotyčnou posloupnost prostě dosadíme do dané rovnice a uvidíme, které konstanty povedou k jejímu splnění.

Chceme tedy, aby pro všechna $n\geq 0$ platilo

$$\begin{aligned}
a_{n+2}-2a_{n+1}-3a_n &= -9n2^n\\
(A(n+2)+B)2^{n+2}-2(A(n+1)+B)2^{n+1}-3(An+B)2^n &= -9n2^n\\
(An+2A+B)\cdot 4-2(An+A+B)\cdot 2-3(An+B) &= -9n\\
4An+8A+4B-4An-4A-4B-3An-3B &= -9n\\
-3An+4A-3B &= -9n\\
[-3A]n+[4A-3B] &= -9n+0.
\end{aligned}$$

Dostali jsme rovnost dvou polynomů a víme, že dva polynomy se rovnají, jen pokud se rovnají jejich koeficienty (to vyplývá z toho, že jednotlivé mocniny jsou jako funkce lineárně nezávislé). Proto máme rovnice $-3A=-9$ a $4A-3B=0$, odkud obratem ruky dostaneme $A=3$, $B=4$. Právě jsme zjistili, že daná rovnice má řešení $\{(3n+4)2^n\}$.

Protože díky větě o struktuře řešení víme, že obecné řešení se dá získat jako jedno partikulární plus obecné homogenní, můžeme napsat odpověď: Obecné řešení dané rovnice je $\{(3n+4)2^n+(-1)^nu+3^nv\}_{n=0}^{\infty}$.

△

Toto je typický průběh řešení lineární rekurentní rovnice s konstantními koeficienty a kvazipolynomiální pravou stranou. Všimněte si, že charakteristická čísla závisí čistě na levé straně dané rovnice, takže nám ji popisují, říkají nám, jak se chová. Naopak to číslo $\lambda$ nám popisuje zásadní chování pravé strany (polynom už ji jen modifikuje). Jestliže se $\lambda$ a charakteristická čísla nepřekrývají, pak to znamená, že se levá a pravá strana neovlivňují a řešení dostaneme v jednodušším tvaru. Pokud by se ale $\lambda$ rovnala některému charakteristickému číslu, pak to ukazuje, že levá a pravá strana mají něco společného, můžeme si představit, že spolu rezonují, že mezi nimi existuje nějaká vazba. To se pak musí projevit v našem odhadnutém řešení, přidáváme tam násobící faktor $n$, a to tolikrát, kolik je násobnost $\lambda$, čili jak moc se ty dvě strany rovnice ovlivňují. Dá se říci, že to $n^m$ je korekční faktor pro případy vzájemného ovlivnění levé a pravé strany.

**!** **Příklad 10b.f:** Najdeme řešení rovnice $a_n=2a_{n-1}+3\cdot 2^n$ pro $n\geq 1$ s podmínkou $a_0=13$.

Nejprve si ji přepíšeme tak, aby neznámé byly na levé straně: $a_n-2a_{n-1}=3\cdot 2^n$. Teď ji přepíšeme do standardního tvaru, tedy zvětšíme všechna $n$ o jedničku: $a_{n+1}-2a_n=3\cdot 2^{n+1}$. Nejmenší $a_i$ zmíněné v původní rovnici pro $n=1$ bylo $a_0$, aby to platilo i pro naši přepsanou rovnici, musíme brát $n\geq 0$.

Pokud dáváte přednost formálnímu přístupu, pak v rovnici $a_n-2a_{n-1}=3\cdot 2^n$ použijte substituci $m=n-1$, dostanete $n=m+1$ a tedy $a_{m+1}-2a_m=3\cdot 2^{m+1}$. Tato rovnice má být platná pro $n\geq 1$, tedy pro $m+1\geq 1$, což znamená $m\geq 0$.

Každopádně máme lineární rekurentní rovnici s konstantními koeficienty $a_{n+1}-2a_n=3\cdot 2^{n+1}$, $n\geq 0$ a nejprve budeme hledat obecné řešení. Jako obvykle začneme přidruženou homogenní rovnicí.

1) Rovnice $a_{n+1}-2a_n=0$ má charakteristický polynom $p(\lambda)=\lambda-2$ a tudíž charakteristické číslo $\lambda=2$. Dostáváme obecné „homogenní řešení“ $a_{h,n}=2^nu$ pro $u\in\mathbb{R}$.

2) Teď potřebujeme najít nějaké řešení rovnice $a_{n+1}-2a_n=6\cdot 2^n$, pravou stranu jsme si přepsali, aby bylo vidět, že je to kvazipolynom. Jeho parametry jsou $\lambda=2$ a $P(n)=6$. Hned vidíme, že toto $\lambda$ se překrývá s charakteristickými čísly levé strany, jmenovitě jednou, proto $m=1$ a v námi uhodnutém řešení bude muset být opravný člen. Stupeň polynomu $P$ je nula, proto podle Věty 10b.7 musí existovat polynom $Q$ stupně 0 takový, že $\{n^1Q(n)2^n\}$ je řešení naší rovnice. Obecný polynom nultého stupně má tvar $Q(n)=A$ pro nějaké $A\in\mathbb{R}$, dostáváme tedy následující uhodnuté řešení: $a_n=An\,2^n$.

Zbývá najít $A$, takže dosadíme toto odhadnuté řešení do rovnice a uvidíme, jaké $A$ bude fungovat:

$$\begin{aligned} a_{n+1} - 2a_n &= 6 \cdot 2^n \\ A(n+1)2^{n+1} - 2 \cdot An2^n &= 6 \cdot 2^n \\ 2A(n+1) - 2An &= 6 \\ 2A &= 6 \implies A = 3. \end{aligned}$$

Máme tedy žádané řešení, je to $a_{p,n} = 3n\,2^n$. Když jej přičteme k $a_{h,n}$, dostaneme obecné řešení.

Závěr: Obecné řešení rovnice je $\{3n\,2^n + 2^n u\}_{n=0}^{\infty}$ pro $u \in \mathbb{R}$.

Protože jsme přeci jen řešili trochu jinou rovnici než tu původní, raději uděláme zkoušku pro rovnici ze zadání: Nejmenší hodnota $n$, která se v zadané rovnici vyskytuje, je $n-1$ pro $n=1$, což je $n=0$. Naše posloupnost tedy začíná správným indexem. Teď dosadíme, začneme komplikovanější pravou stranou:

$$\begin{aligned} 2a_{n-1} + 3 \cdot 2^n &= 2(3(n-1)\,2^{n-1} + 2^{n-1}u) + 3 \cdot 2^n = 3(n-1)2^n + 2^n u + 3 \cdot 2^n \\ &= 3 \cdot (n-1+1)2^n + 2^n u = 3n\,2^n + 2^n u = a_n. \end{aligned}$$

Rovnice je splněna.

3) Teď najdeme partikulární řešení, které splňuje danou počáteční podmínku:

$$13 = a_0 = 3 \cdot 0 \cdot 2^0 + 2^0 u = 0 + u \implies u = 13.$$

Zadaná úloha má řešení $\{(3n+13)2^n\}_{n=0}^{\infty}$.

Poznámka: Co by se stalo, kdybychom zapomněli na opravný faktor? Mysleli bychom si naivně, že existuje řešení ve tvaru $a_n = A \cdot 2^n$. Po dosazení do rovnice bychom dostali

$$\begin{aligned} a_{n+1} - 2a_n &= 6 \cdot 2^n \\ A2^{n+1} - 2 \cdot A2^n &= 6 \cdot 2^n \\ 2A - 2A &= 6 \\ 0 &= 6. \end{aligned}$$

A máme smůlu. Takže ty opravné faktory opravdu k něčemu jsou.

△

Shrneme si postup.

**S** **Algoritmus 10b.8.** pro nalezení řešení rovnice $a_{n+k} + c_{k-1}a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = b_n$ pro všechna $n \geq n_0$, kde $b_n = P(n)\lambda^n$, $c_i \in \mathbb{R}$ a $c_0 \neq 0$ (tedy řád $k$).

**1.** Nejprve řešte přidruženou homogenní rovnici $a_{n+k} + c_{k-1}a_{n+k-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0$.

a) Najděte všechna charakteristická čísla $\lambda_j$ s násobnostmi $m_j$ řešením rovnice $\lambda_k + c_{k-1}\lambda^{k-1} + \cdots + c_1\lambda + c_0 = 0$.

b) Podle Algoritmu 10b.5 sestavte bázi prostoru řešení $B = \big\{\{a_{i,n}\}_{n=n_0}^{\infty};\ i = 1, \dots, k\big\}$.

c) Obecné řešení přidružené homogenní rovnice je $\{a_{h,n}\} = \left\{\sum\limits_{i=1}^{k} u_i a_{i,n}\right\}$ pro $u_i \in \mathbb{R}$.

Pokud byla zadaná rovnice již homogenní, jděte na **3**.

**2.** Pokud nebyla zadaná rovnice homogenní, zkontrolujte, že je pravá strana kvazipolynomem, tedy $b_n = P(n)\lambda^n$ pro nějaké $\lambda \in \mathbb{R}$ a polynom $P$.

a) Porovnejte $\lambda$ s charakteristickými čísly $\lambda_j$ z kroku 1. Pokud se žádnému nerovná, položte $m = 0$. Pokud pro nějaké $j$ platí $\lambda = \lambda_j$, položte $m = m_j$ (násobnost dotyčného charakteristického čísla).

b) Sestavte obecný polynom $Q$ stupně stejného jako $P$, tradičně se používá $Q(n) = A + Bn + \cdots$.

c) Uhádněte řešení $a_n = n^m Q(n)\lambda^n$. Dosaďte jej do dané rovnice a po zkrácení $\lambda$ zjednodušte levou stranu do tvaru polynomu. Porovnáním koeficientů polynomů na levé a pravé straně získáte tolik rovnic, kolik je neznámých koeficientů v $Q$.

d) Vyřešte tyto rovnice a obdržené konstanty dosaďte zpět do $Q$. Získáte jedno konkrétní řešení $a_{p,n}$.

e) Obecné řešení dané úlohy je $\left\{a_{p,n} + \sum\limits_{i=1}^{k} u_i a_{i,n}\right\}_{n=n_0}^{\infty}$ či $a_n = a_{p,n} + \sum\limits_{i=1}^{k} u_i a_{i,n}$ pro $n \geq n_0$.

**3.** Pokud byly s rovnicí zadány také počáteční podmínky, dosaďte za $a_j$ v těchto podmínkách vzorce pro $a_j$ z obecného řešení, které jste našli. Získáte $k$ rovnic pro $k$ neznámých $u_1, \dots, u_k$. Vyřešte tuto soustavu, získaná $u_i$ dosaďte do vzorce pro obecné řešení a dostanete tak partikulární řešení pro zadanou úlohu.

△

**S** Mechanismus vzniku odhadu je vlastně velice snadný. Geometrický faktor $\lambda^n$ opíšeme, polynom také, jen jej změníme na obecný. V případě shody $\lambda$ na pravé a levé straně ještě přidáme korekční faktor. Vyzkoušíme to v následující tabulce. Ve sloupcích jsou levé strany rovnic a v řádcích napravo jsou pravé strany. Například druhé pole zleva v prvním řádku odpovídá rovnici $a_{n+2} - 3a_{n+1} + 2a_n = n2^n$, vyplnili jsme příslušný odhad.

| | $a_{n+2}-9a_n=$ $[\lambda=-3,3]$ | $a_{n+2}-3a_{n+1}+2a_n=$ $[\lambda=1,2]$ | $a_{n+2}-4a_{n+1}+4a_n=$ $[\lambda=2\ (2\times)]$ | $L=\big/=b_n$ |
|---|---|---|---|---|
| | $(An+B)2^n$ | $n(An+B)2^n$ | $n^2(An+B)2^n$ | $=n\,2^n$ $[\lambda=2]$ |
| | $(An^2+Bn+C)(-1)^n$ | $(An^2+Bn+C)(-1)^n$ | $(An^2+Bn+C)(-1)^n$ | $=n^2(-1)^n$ $[\lambda=-1]$ |
| | $An+B$ | $n(An+B)$ | $An+B$ | $=2n-5$ $[\lambda=1]$ |
| | $nA(-3)^n$ | $A(-3)^n$ | $A(-3)^n$ | $=(-3)^n$ $[\lambda=-3]$ |

Je dobré se na to podívat po řádcích. V prvním řádku krásně vidíme, jak se vždy v odhadech řešení zachovává geometrická posloupnost a zobecněný polynom z pravé strany, někdy pak přidáváme opravný faktor podle toho, kolikrát se $\lambda$ napravo najde také jako charakteristické číslo levé strany. Zajímavý je třetí řádek, tam žádný geometrický člen nebyl a toto se také v odhadech zachovalo, při porovnávání je pak třeba použít $\lambda=1$, protože pravou stranu lze přepsat jako $(2n-5)\cdot 1^n$. Podobně v posledním řádku máme vlastně $1\cdot(-3)^n$, tudíž je tam jakoby polynom stupně nula, což se zobecní jako konstanta.

Ukážeme teď jeden zajímavý příklad a pak si zkusíme působnost tohoto algoritmu trochu rozšířit.

**! Příklad 10b.g:** Zde se podíváme, kolik operací „stojí“ výpočet determinantu matice o rozměru $n\times n$.

**a)** Podle definice se mají sčítat součiny typu $a_{1\pi(1)}\cdot a_{2\pi(2)}\cdots a_{n\pi(n)}$, kde suma jde přes všechny permutace $\pi$ množiny $\{1,2,\dots,n\}$. Těchto permutací je $n!$ a jeden takový součin stojí $n-1$ násobení plus jedno přičtení k ostatním, tedy výpočet determinantu podle definice vyžaduje celkem $n\cdot n!$ operací, a to jsme ještě nezapočítali vytváření těch premutací. Jde tedy o zcela neperspektivní způsob, jakmile se mohou objevit trochu větší matice.

My takto počítáme determinanty $2\times 2$ a $3\times 3$ tužkou na papír, kdy to ještě jde. Pro trochu větší matice se s oblibou používá rozvoj podle řádku či sloupce, který je případně opakován, dokud se nedojde k malým determinantům. Byla by tato metoda dobrým východiskem pro algoritmus?

**b)** Nechť $a_n$ je náročnost počítání determinantu rozvojem podle prvního sloupce, je-li velikost matice $n\times n$. K jeho spočítání si nejprve musíme spočítat $n$ doplňkových determinantů, které mají o jedno menší velikost, takže na to potřebujeme $n\cdot a_{n-1}$ operací. Ty pak potřebujeme střídavě přičítat a odčítat, dostaneme $a_n=n\cdot a_{n-1}+n$ operací. Pro matici $1\times 1$ samozřejmě potřebujeme $a_1=1$, což je počáteční podmínka.

Když to přepíšeme, máme $a_{n+1}-(n+1)a_n=n+1$, což je sice krásná lineární rekurentní rovnice, ale bohužel nemá konstantní koeficienty, takže jsme se ji vyřešit nenaučili. To ale nakonec nebude tak velký problém, my teď totiž ukážeme, že i tento algoritmus je natolik náročný, že nás nebude zajímat.

Dokážeme indukcí, že každé řešení $a_n$ této rovnice splňuje $a_n\geq n!$.

(0) $a_1=1=1!$, v pořádku.

(1) Předpokládejme, že $a_n\geq n!$. Pak $a_{n+1}=(n+1)a_n+n+1\geq(n+1)n!+n+1\geq(n+1)!$.

Oproti výpočtu podle definice jsme si tedy moc nepolepšili a jako obecná metoda to není vhodný nápad, nicméně stojí za zmínku, že pokud dopředu víme, že hodně členů matice je nulových, pak se náročnost prudce snižuje a rozvoj podle sloupce/řádku se stává zajímavou volbou, viz cvičení 10b.7.

**c)** Zkusíme teď jiný postup, uděláme zase rozvoj podle prvního sloupce, ale tentokráte si nejdříve vyrobíme nuly pod levým horním rohem, abychom pak nemuseli počítat tolik subdeterminantů. Kolik operací bude stát takovýto výpočet determinantu matice $n\times n$, označme to $d_n$?

Zkusme totálně pesimistickou variantu. Vlevo nahoře může být nula. Pak nejprve projdeme ostatní členy v prvním sloupci, abychom našli něco nenulového (pesimisticky $n$ operací, bude to až dole), pak prohodíme první a poslední řádek ($n$ operací, musíme prohodit všechny členy v řádcích) a někde si zapamatujeme, že tím pádem počítáme mínus ten determinant. Příprava prvního řádku nás tedy pesimisticky stála $2n$ operací. No a pak odečítáme příslušný násobek prvního řádku od ostatních, každý řádek nás stojí 2 operace na člen (násobení, odečtení), tedy zase $2n$ operací na řádek, musíme jich takto upravit $n-1$. Sečteno: vyžaduje to $2n+2n(n-1)=2n^2$ operací.

Teď je tedy matice připravena a rozvíjíme podle členu vlevo nahoře, takže musíme spočítat jeden determinant matice o řád menší a vynásobit jej tím levým horním členem. Dostáváme tak rovnici $d_n=d_{n-1}+2n^2+1$. Opět si ji přepíšeme: Chceme vyřešit $d_{n+1}-d_n=2(n+1)^2+1$ s počáteční podmínkou $d_1=1$.

Toto už je lineární rekurentní rovnice s konstantními koeficienty, takže nasadíme standardní metody. Nejprve homogenní rovnice: $d_{n+1}-d_n=0$ dává charakteristický polynom $\lambda-1$ a tudíž vychází charakteristické číslo $\lambda=1$. Obecné řešení této homogenní rovnice je konstantní posloupnost $\{u\cdot 1^n\}_{n=1}^{\infty}=\{u\}_{n=1}^{\infty}$.

Teď potřebujeme partikulární řešení, přepis $d_{n+1} - d_n = 2n^2 + 4n + 3 = (2n^2 + 4n + 3)1^n$ ukazuje, že máme kvazipolynomiální pravou stranu, která má $\lambda = 1$, což se shoduje s charakteristickým číslem (jednoduchým) levé strany. V námi odhadnutém řešení tedy bude muset být opravný člen $n^1$ a také obecný polynom $Q$ stupně 2. Odhadneme proto řešení ve tvaru $d_n = n^1(An^2 + Bn + C)1^n = An^3 + Bn^2 + Cn$. Dosadíme jej do dané rovnice a levou stranu pak upravíme na polynom:

$$\begin{aligned} d_{n+1} - d_n &= 2n^2 + 4n + 3 \\ (A(n+1)^3 + B(n+1)^2 + C(n+1)) - (An^3 + Bn^2 + Cn) &= 2n^2 + 4n + 3 \\ An^3 + A3n^2 + A3n + A + Bn^2 + B2n + B + Cn + C - An^3 - Bn^2 - Cn &= 2n^2 + 4n + 3 \\ (3A)n^2 + (3A + 2B)n + (A + B + C) &= 2n^2 + 4n + 3. \end{aligned}$$

Porovnáním koeficientů u stejných mocnin na levé a pravé straně dostáváme soustavu rovnic $3A = 2$, $3A+2B = 4$, $A + B + C = 3$, z ní hravě odvodíme $A = \frac{2}{3}$, $B = 1$, $C = \frac{4}{3}$.

Dostáváme tedy partikulární řešení $\bigl\{\frac{2}{3}n^3 + n^2 + \frac{4}{3}n\bigr\}$ a obecné řešení $\bigl\{\frac{2}{3}n^3 + n^2 + \frac{4}{3}n + u\bigr\}$. My ovšem hledáme řešení splňující počáteční podmínku $d_1 = 1$, takže chceme $\frac{2}{3} + 1 + \frac{4}{3} + u = 1$, což dává $u = -2$.

Závěr: Daná rovnice má řešení $\bigl\{\frac{2}{3}n^3 + n^2 + \frac{1}{3}n - 2\bigr\}_{n=1}^{\infty}$.

Popravdě řečeno, většinu práce jsme si mohli odpustit. Nás u algoritmů zajímá jejich asymptotická rychlost, přičemž vidíme, že homogenní řešení jsou konstantní posloupnosti, takže růst řešení určí spíš ten odhad. V něm byl dominantním členem $n^3$, to už jsme viděli i předtím, než jsme konstanty našli (protože $A$ v takových případech vyjde nenulové). Nemuseli jsme tedy konstanty nacházet, už v okamžiku odhadu bylo vidět, že studovaný algoritmus potřebuje řádově $n^3$ operací neboli $d_n = \Theta(n^3)$.

Když si představíte, jak tento algoritmus funguje rekurzivně přes celou matici, tak byste měli dospět k názoru, že to vlastně není nic jiného, než že matici upravíme na trojúhelníkový tvar a pak vynásobíme členy na diagonále. Je to proto úzce svázáno s Gaussovou eliminací a o té je známo, že potřebuje řádově $n^3$ operací. Náš výpočet tedy dal správnou odpověď. Náročnost $n^3$ je upřímně řečeno také dost drsná, ale oproti faktoriálu je to totální revoluce.
$\triangle$

Existují pravé strany, které jsou velice blízké kvazipolynomům. Například $b_n = n \cdot 3^n + 7 \cdot 2^n$ vypadá hodně jako kvazipolynom, ale tento výraz nelze sjednotit pomocí jediného $\lambda^n$, je jich tam víc. Zde pomůže princip, který je velice obecný a zase jej najdeme u všech typů lineárních rovnic.

! **Věta 10b.9.** (o superpozici)
Nechť $k \in \mathbb{N}$, uvažujme funkce $c_0(n), c_1(n), \dots, c_{k-1}(n)\colon \mathbb{Z} \mapsto \mathbb{R}$.

Jestliže posloupnost $\{a_n\}_{n=n_0}^{\infty}$ řeší rovnici $\quad a_{n+k} + \sum\limits_{i=0}^{k-1} c_i(n) a_{n+i} = b_n \quad$ pro všechna $n \geq n_0$

a posloupnost $\{\tilde{a}_n\}_{n=n_0}^{\infty}$ řeší rovnici $\quad a_{n+k} + \sum\limits_{i=0}^{k-1} c_i(n) a_{n+i} = \tilde{b}_n \quad$ pro všechna $n \geq n_0$,

pak posloupnost $\{a_n + \tilde{a}_n\}_{n=n_0}^{\infty}$ řeší rovnici $\quad a_{n+k} + \sum\limits_{i=0}^{k-1} c_i(n) a_{n+i} = b_n + \tilde{b}_n \quad$ pro všechna $n \geq n_0$.

Důkaz je snadný a necháme jej jako cvičení 10b.8.

Indukcí se to dá snadno rozšířit na libovolný (konečný) počet sčítanců na pravé straně. Věty o superpozici jsou jakýmsi evergreenem všech oblastí matematiky zabývajícími se lineárními rovnicemi a umožňují nám rozdělit nepříjemnou pravou stranu na příjemnější sčítance, rovnice pak řešíme pro každý z nich zvlášť.

V našem případě jde o součet kvazipolynomů. Pak nemusíme se sčítámím jednotlivých částečných řešení čekat až na konec postupu, ale lze dávat dohromady již jednotlivé odhady. Ukážeme si to na příkladě.

! **Příklad 10b.h:** Vyřešíme rovnici $a_{n+2} - a_n = 3 \cdot 2^n - 12$ pro $n \geq 0$ s počátečními podmínkami $a_0 = 1$, $a_1 = 0$.

Řešení: Jde o lineární rekurentní rovnici s konstantními koeficienty, takže aplikujeme standardní algoritmus. Nejprve řešíme homogenní rovnici: $a_{n+2} - a_n = 0$ dává charakteristickou rovnici $\lambda^2 - 1 = 0$, jsou tedy jednoduchá charakteristická čísla $\lambda = \pm 1$ a obecné řešení $\{u \cdot (-1)^n + v \cdot 1^n\}_{n=0}^{\infty} = \{(-1)^n u + v\}_{n=0}^{\infty}$.

Teď potřebujeme nějaké partikulární řešení. Bohužel pravá strana není kvazipolynomiální, protože se nedá napsat jako polynom krát nějaké $\lambda^n$. Je to ale součet dvou kvazipolynomů, $3 \cdot 2^n$ a $-12 = -12 \cdot 1^n$. Pro každý z nich teď odhadneme řešení.

Kvazipolynom $3 \cdot 2^n$ má polynom $P(n) = 3$ stupně 0, budeme tedy potřebovat $Q(n) = A$, a speciální číslo $\lambda = 2$, které nepatří mezi charakteristická čísla levé strany, takže není třeba dělat korekci, $m = 0$. Proto uhodneme řešení $A2^n$.

Kvazipolynom $-12 \cdot 1^n$ má polynom $P(n) = -12$ stupně 0, budeme potřebovat $Q(n) = B$ (zde jsme použili jiné písmeno, abychom mohli vznikající odhad přičíst k předchozímu), a speciální číslo $\lambda = 1$, které je charakteristickým číslem levé strany, a to jednou, bude tedy korekce s $m = 1$. Proto uhodneme řešení $n^1 \cdot B \cdot 1^n = Bn$.

Pro celou danou pravou stranu tak dostáváme řešení ve tvaru $a_n = A2^n + Bn$. Konstanty $A, B$ určíme dosazením našeho odhadu do dané rovnice:

$$\begin{aligned} a_{n+2} - a_n &= 3 \cdot 2^n - 12 \\ (A2^{n+2} + B(n+2)) - (A2^n + Bn) &= 3 \cdot 2^n - 12 \\ 4A2^n + Bn + 2B - A2^n - Bn &= 3 \cdot 2^n - 12 \\ 3A2^n + 2B &= 3 \cdot 2^n - 12. \end{aligned}$$

Na rozdíl od předchozích příkladů se nám teď nepodařilo zkrátit $\lambda^n$, což není překvapení, protože vlastně máme v úloze dvě různé lambdy. Není to ale žádný problém, mocniny $2^n$ a $1^n$ jsou lineárně nezávislé, takže jen zobecníme předchozí metodu. Budeme navzájem porovnávat koeficienty na levé a pravé straně u stejných mocnin $n^i$ a $\lambda^n$. Vidíme tam mocninu $n^0 2^n$, na levé straně je u ní $3A$ a na pravé straně je u ní 3. Proto $A = 1$. Podobně vidíme na levé straně u mocniny $n^0 1^n$ koeficient $2B$ a na pravé straně $-12$, proto $B = -6$. Tyto konstanty dosadíme do našeho odhadnutého řešení a nalezli jsme partikulární řešení $\{2^n - 6n\}$.

Teď použijeme větu o struktuře řešení, obecné řešení dané rovnice je $\{2^n - 6n + (-1)^n u + v\}_{n=0}^{\infty}$.

Zbývá zpracovat počáteční podmínky:

$$\begin{aligned} a_0 &= 1 + u + v = 1 \\ a_1 &= 2 - 6 - u + v = 0 \end{aligned} \implies \begin{aligned} u + v &= 0 \\ -u + v &= 4 \end{aligned} \implies u = -2, v = 2.$$

Řešení úlohy je $\{2^n - 6n - 2 \cdot (-1)^n + 2\}_{n=0}^{\infty}$.

$\triangle$

Všimněte si, že větu 10b.9 jsme formulovali pro všechny lineární rovnice, nejen ty s konstantními koeficienty, takže jsme ji mohli zařadit do kapitoly 10a. Dokonce by trochu pomohla, protože jsme při důkazu Věty 10a.4 používali speciální případ superpozice, kdy je jedna pravá strana nulová. Čtenář by ale asi nevěděl, odkud se tam bere a co s ní, tady je na správném místě.

**Příklad 10b.i** (pokračování 10a.b)**:** Vraťme se ke králíkům. Odhadli jsme, že poté, co se králíci deset let nerušeně množili, bylo jich v roce 1869 cca $24 \cdot 10^6$. Podíváme se na dva scénáře. Označme jako $a_n$ počet králíků $n$ let po roce 1869, a to v milionech (ušetříme šest nul).

**a)** Pokud by se králíci nerušeně množili dál, pak by jejich populace zase splňovala rovnici $a_{n+1} = 4a_n$ neboli $a_{n+1} - 4a_n = 0$ a počáteční podmínku $a_0 = 24$.

Tato homogenní rovnice má charakteristické číslo $\lambda = 4$ a obecné řešení $a_n = u \cdot 4^n$, počáteční podmínka pak dává $a_n = 24 \cdot 4^n$.

**b)** Druhý scénář využívá informace, že se střílelo $2 \cdot 10^6$ králíků ročně. Můžeme si to interpretovat tak, že je vystřílíme na konci roku, což dává upravený vztah $a_{n+1} = 4a_n - 2$, tedy $a_{n+1} - 4a_n = -2$, zase s počáteční podmínkou $a_0 = 24$.

Toto je standardní lineární rekurentní rovnice, u níž nejprve najdeme řešení přidružené homogenní úlohy. To už jsme udělali ve scénáři a), dostali jsme $a_{h,n} = 4^n u$.

Teď potřebujeme nějaké partikulární řešení. Pravá strana $-2 = -2 \cdot 1^n$ je kvazipolynom, který má polynom $P(n) = -2$ stupně 0, proto použijeme $Q(n) = A$, a číslo $\lambda = 1$, které není stejné jako charakteristické číslo levé strany. Proto $m = 0$ a korekce nebude třeba. Dostáváme odhadnuté řešení $a_n = A1^n = A$. Správné $A$ najdeme dosazením $a_n$ do dané rovnice:

$$\begin{aligned} a_{n+1} - 4a_n &= -2 \\ A - 4A &= -2 \\ -3A &= -2 \implies A = \tfrac{2}{3}. \end{aligned}$$

Máme tedy obecné řešení naší rovnice ve tvaru $a_n = \frac{2}{3} + 4^n u$. Když použijeme počáteční podmínku $a_0 = 24$, dostaneme $\frac{2}{3} + u = 24$, tedy $u = 23 + \frac{1}{3}$.

Druhý scénář dává růst populace $a_n = \frac{2}{3} + \left(23 + \frac{1}{3}\right)4^n$. Vidíme, že populace roste pomaleji, ale pořád rychlostí geometrické posloupnosti $4^n$, takže k nějaké zásadnější změně nedošlo.

Zajímavá úloha: Zkuste vyřešit stejnou úlohu, ale teď s úbytkem $L$, tedy vyřešte rovnici $a_{n+1} - 4a_n = -L$ s podmínkou $a_0 = 24$. Stejným postupem dostanete vzorec, ve kterém se objeví parametr $L$, a položte si otázku, jak velké by $L$ muselo být, aby již $a_n$ nerostlo rychlostí $4^n$ (nápověda: vzorec vyjde $a_n = \frac{L}{3} + \left(24 - \frac{L}{3}\right)4^n$). Je vidět, že byste museli na konci každého roku vystřílet 72 milionů králíků. Možná vás napadne, že chyba je v tom, že je střílíme až na konci roku, když se rozmnoží. Bohužel si člověk nepomůže ani střílením třeba hned na začátku roku. Pak je správná rovnice dána $a_{n+1} = 4(a_n - L) = 4a_n - 4L$, je to tedy vlastně původní příklad, jen se efekt $L$ čtyřikrát zvětší. Na zabránění geometrického růstu by bylo proto třeba střílet minimálně $\frac{72}{4} = 18$ miliónů ročně, což je pořád nerealistické.

Mnohem perspektivnější je změnit tu čtyřku neboli rozmnožovací konstantu, což je v zásadě základ moderních metod boje proti škůdcům.

Poznámka: Problém s tím, kdy králíky střílíme, se dá řešit tak, že pracujeme s menšími časovými intervaly. Můžeme například uvažovat stav po týdnech $b_n$, dostáváme pak rovnici $b_{n+1} = Kb_n - L$, kde $L$ je hodnota týdenního odstřelu a $K$ je konstanta přirozené množivosti, která samozřejmě souvisí s tou čtyřkou za rok. Pokud bychom nestříleli, bylo by za 52 týdnů $K^{52}$ krát více králíků, proto musí platit $K^{52} = 4$. Odtud již $K$ snadno určíme, rovnici lze vyřešit a roční stavy dostaneme jednoduchým převodem $a_n = b_{52n}$. Tento model je již mnohem realističtější. Lze dokonce počítat i stav po dnech, pak už je ale lepší přejít rovnou na diferenciální rovnice, kdy měříme čas na reálné ose, čímž se dostáváme mimo oblast diskrétní matematiky, toto je hájemství matematiky spojité neboli matematické analýzy.

△

**!** **10b.10 Poznámka:** Onen první model neomezeného růstu králíků daný vztahem $a_{n+1} = Ka_n$ patří přes svou jednoduchost k jedněm z nejzajímavějších a nejpoužívanějších modelů, říká se mu exponenciální model, protože vede na řešení $c \cdot K^n = c \cdot e^{\ln(K)n}$. I když často vykazuje jen přibližnou shodu s reálným jevem, lze jej použít k rychlému orientačnímu náhledu, což bývá velice užitečné. Některé populární situace:

• Máte na účtě $a_0 = C$ peněz, každý rok se vám to zúročí hodnotou $r$ procent. Pak vývoj financí popisuje (dokonce přesně) vztah $a_{n+1} = \bigl(1 + \frac{r}{100}\bigr)a_n$.

• Máte-li na začátku kolonii $a_0 = C$ bakterií v Petriho misce, pak jejich počet po jednotlivých hodinách je dán vztahem $a_{n+1} = Ka_n$. Výsledná exponenciála vykazuje dobrou shodu s realitou, dokud není počet natolik velký, že baktérie začnou soupeřit o zdroje živin.

• Totéž lze říci o libovolné populaci, která se volně množí, neužírají z ní predátoři a má dostatek zdrojů. I lidstvo se tak jeden čas porůznu množilo.

• Když se rozpadne atom uranu 235, vypustí dva neutrony. Když neutron vhodně vletí do dalšího atomu uranu, ten se rozpadne a vypustí dva neutrony. Počet rozpadlých jader po $n$ krocích je dán $a_{n+1} = 2a_n$, což vede na $2^n$ a říká se tomu jaderný výbuch. Ve skutečnosti je počet vypuštěných neutronů proměnný, udává se v průměru okolo 2.5, ale ten násobící faktor v rovnici bývá menší, protože určité procento neutronů se netrefí do dalšího atomu a uletí bez užitku pryč. V jaderných elektrárnách se toto snaží podporovat, dokonce neutrony i lapají. Tento příklad je samozřejmě zjednodušený, ale vystihuje podstatu.

• Radioaktivní prvky fungují tak, že se jednotlivé atomy neustále náhodně rozhodují, zda se rozpadnou nebo ne. Jestliže máme $a_n$ atomů, tak se z nich během sekundy určité procento $p$ rozpadne, takže pak už jich máme $a_{n+1} = \bigl(1 - \frac{p}{100}\bigr)a_n$. Vede to na řešení $a_n = c\bigl(1 - \frac{p}{100}\bigr)^n$, které platí i pro hmotnost či hustotu zastoupení v jiném materiálu a velice dobře odpovídá reálnému stavu (většinou se ale zase pracuje se spojitým časem a vyjde exponenciála $c\,e^{\lambda t}$). Na tomto vzorci je mimo jiné založena metoda měření stáří organických látek ve vykopávkách.

△

**!** **Příklad 10b.j:** V kapitole o indukci jsme si dokazovali rozličné součtové vzorce (viz např. cvičení 5a.1), ale poznamenali jsme, že na to, abychom dokázali indukcí jejich správnost, je musíme odněkud získat. Jedna možnost byla naznačena v důkazu Věty 9c.3, zde si ukážeme další.

Najdeme vzorec pro $\sum\limits_{k=1}^{n} k = 1 + 2 + \cdots + n$.

Hledanou hodnotu si označíme $s_n = \sum\limits_{k=1}^{n} k$. Pak platí rekurentní vztah $s_{n+1} = s_n + (n+1)$, máme tedy rovnici $s_{n+1} - s_n = n + 1$, evidentně platí počáteční podmínka $s_1 = 1$.

Tuto úlohu vyřešíme standardním způsobem. Nejprve najdeme obecné řešení přidružené homogenní rovnice $s_{n+1} - s_n = 0$. Charakteristická rovnice $\lambda - 1 = 0$ dává charakteristické číslo $\lambda = 1$ a konstantní posloupnost $\{u \cdot 1^n\}_{n=1}^{\infty} = \{u\}_{n=1}^{\infty}$ jako obecné řešení homogenní rovnice.

Teď najdeme jedno partikulární řešení. Pravá strana je kvazipolynomiální, $n + 1 = (n+1)1^n$, proto lze použít metodu odhadu. Zde je speciální číslo $\lambda = 1$ také charakteristickým číslem (jednoduchým), máme tedy korekční faktor $m = 1$ a odhadneme řešení $s_n = n^1(An + B)1^n = An^2 + Bn$. Dosadíme do řešené rovnice, upravíme na polynom a porovnáme koeficienty:

$$s_{n+1}-s_n=n+1$$
$$(A(n+1)^2+B(n+1))-(An^2+Bn)=n+1$$
$$2An+(A+B)=n+1 \implies 2A=1,\ A+B=1 \implies A=B=\tfrac{1}{2}.$$

Máme tedy partikulární řešení $\bigl\{\frac{1}{2}n^2+\frac{1}{2}n\bigr\}$ a obecné řešení $\bigl\{\frac{1}{2}n^2+\frac{1}{2}n+u\bigr\}_{n=1}^{\infty}$. Teď ještě najdeme hodnotu $u$ tak, aby tato posloupnost splňovala počáteční podmínku:

$$s_1=\tfrac{1}{2}+\tfrac{1}{2}+u=1 \implies u=0.$$

Vychází $s_n=\frac{1}{2}n^2+\frac{1}{2}n=\frac{n(n+1)}{2}$, což je onen známý vzorec.
△

**Příklad 10b.k:** Máme k dispozici dva různé signály lišící se délkou (pípnutí, třeba jako u Morseovky), jeden trvá 1 ms, druhý 2 ms. Kolik různých zpráv se dá poslat za $n$ ms?

Toto je kombinatorická úloha, kterými se zabýváme v kapitole 11b, je to tedy jakási upoutávka na to, co přijde, a zároveň připomínka, že v dotyčné kapitole čtenář najde další příklady s rekurentními vztahy.

Označme jako $a_n$ počet různých zpráv, které lze vyslat za $n$ milisekund. Uvažujme množinu všech takových zpráv. Tato množina se dá rozdělit na dvě části podle toho, jestli ve zprávě je jako poslední krátký či dlouhý signál. Počet zpráv, které končí krátkým signálem, je stejný, jako počet zpráv o délce $n-1$. Počet zpráv, které končí dlouhým signálem, je stejný jako počet zpráv o délce $n-2$. Dostáváme tedy rovnici $a_n=a_{n-1}+a_{n-2}$ pro $n\geq 2$. Je to zase Fibonacciho vztah!

Jaké jsou počáteční podmínky? Je jen jedna zpráva dlouhá 1 ms, $a_1=1$, zato do dvou ms se vejde buď jeden dlouhý signál nebo dva krátké, tedy dvě různé zprávy, $a_2=2$.

Když si rovnici upravíme, $a_{n+2}-a_{n+1}-a_n=0$, dostáváme homogenní lineární rovnici 2. řádu s konstantními koeficienty, kterou řešíme jako v příkladě 10b.b. Dostáváme obecné řešení $a_n=u\bigl(\frac{1+\sqrt{5}}{2}\bigr)^{n+1}+v\bigl(\frac{1-\sqrt{5}}{2}\bigr)^{n+1}$.

Počáteční podmínky nám pak dají hledané partikulární řešení $a_n=\frac{1}{\sqrt{5}}\bigl(\frac{1+\sqrt{5}}{2}\bigr)^{n+1}-\frac{1}{\sqrt{5}}\bigl(\frac{1-\sqrt{5}}{2}\bigr)^{n+1}$. Je to vlastně posunutá Fibonacciho posloupost, $a_n=F_{n+1}$. Pro ilustraci, je možné vyslat třeba 89 různých zpráv během 10 ms a 987 různých zpráv během 15 ms.
△

**Příklad 10b.l:** Zde si přiblížíme tzv. Josefův problém: Celkem $n$ lidí stojí v kruhu a postupně jsou vyřazováni takto: Jeden se určí jako první, od něj se pak odpočítává a každý $k$-tý člověk vypadává z kruhu. Počítá se pořád dokola stále se zmenšujícího kruhu, dokud nezůstane jeden člověk. Otázka zní: Na jakém místě (v původním rozestavení) stál ten, kdo nakonec zůstal?

O jméno tohoto problému se postaral historik Flavius Josephus. Jednou se prý údajně sešlo v jeskyni 41 poražených vzbouřenců, kteří chtěli tímto způsobem spáchat postupně sebevraždu metodou $k=3$. Mezi nimi byl i Flavius se svým otrokem, ale z nějakého důvodu se nehodlali přidat k ostatním s tou sebevraždou. Vybrali si proto chytře místa tak, aby zůstali jako poslední dva, a pak se nenápadně vytratili. O věrohodnosti této příhody lze oprávněně pochybovat, však to byl historik.

My se podíváme na verzi s $k=2$. Máme $n$ lidí v kruhu očíslovaných od 1 do $n$, rozpočítávají se první druhý a každý se sudým číslem vypadává ze hry, dokud se nedojede na konec kruhu, pak to začne být zajímavé. Protože se to bude hodit, zkusme si to rozmyslet. Začátek kruhu po prvním průchodu vypadá jako $1,3,5,7,\ldots$ Pokud je $n$ sudé, tak jako poslední vypadl pán číslo $n$ a po něm vypadne pán číslo 3. Pokud je liché, tak poslední byl na řadě pán $n-1$, pán $n$ (zatím) zůstává a jako další vypadává 1.

Označíme jako $J(n)$ pořadové číslo toho, kdo nakonec zůstane jako jediný, pokud se začne s $n$ lidmi. Předchozí rozbor nabízí možnost vytvořit rekurentní vztah.

Jestliže byl na začátku sudý počet $n=2m$, pak se po prvním průchodu začne vybírat z lidí $1,3,5,\ldots,(2m-1)$, kterých je $m$, zase metodou každý druhý. Při tom se ale počítá vzhledem k novému pořadí $K$. V jakém vztahu je toto k původnímu pořadí $k$? Ten, kdo je nyní druhý, měl původně číslo 3. Nově třetí měl 5, nově čtvrtý měl 7. Vidíme, že vzorec je $k=2K-1$. Víme vše, co potřebujeme k následující úvaze. Pokud víme, že v probraném kruhu o $m$ lidech nakonec zůstane člověk na pozici $K$, tedy $J(m)=K$, pak tento člověk zůstane i z původních $n=2m$ lidí a měl původně pozici $J(2m)=2K-1$. Dostáváme tedy pro sudý počet lidí rekurentní vztah $J(2m)=2J(m)-1$.

Pokud jsme začali s lichým počtem lidí $n=2m+1$, pak po prvním kole zbydou lidé $1,3,5,7,\ldots,(2m+1)$, kterých je $m+1$. Další vypadne jednička, máme teď $m$ lidí $3,5,7,\ldots,(2m+1)$ a z nich se vybírá metodou každý druhý. Opět potřebujeme najít převodní vztah. Nový první měl číslo 3, nový druhý měl číslo 5 atd, dostáváme $k=2K+1$. Jestliže tedy z nového polovičního kruhu nakonec zůstane člověk na nové pozici $K=J(m)$, pak to je podle původní pozice člověk $J(2m+1)=2K+1$. Pro lichý počet lidí proto dostáváme rekurentní vztah $J(2m+1)=2J(m)+1$. Je to nepříjemné, nemáme jednotnou rovnici pro všechna čísla.

Takovéto situace jsme se řešit nenaučili, nezbývá, než to nějak odhadnout. Jakou hodnotu má $J$ pro malá $n$? $J(1)=1$, $J(2)=1$, $J(3)=3$, $J(4)=1$, $J(5)=3$, $J(6)=5$, $J(7)=7$, $J(8)=1$, $J(9)=3$, $J(10)=5$, $J(11)=7$, $J(12)=9$, $J(13)=11$, $J(14)=13$, $J(15)=15$, $J(16)=1$.

Vidíme, že když je $n$ ve tvaru $2^a$, pak je $J(n)=1$, pak se přidává po dvou. Hypotéza: Jestliže $n=2^a+b$, kde $b<2^a$ a $b\in\mathbb{N}_0$, pak $J(n)=1+2b$. Dokážeme to silnou indukcí.

První počáteční hodnoty se shodují. Teď uděláme indukční krok.

Uvažujme tedy nějaké $n=2^a+b$, kde $b<2^a$ a $b\in\mathbb{N}_0$. Předpokládáme, že vzorec pro $J$ z naší hypotézy je platný pro všechna menší čísla.

Jestliže je $n$ sudé, tedy $n=2m=2^a+b$, pak musí být $b$ sudé a $m=2^{a-1}+\frac{1}{2}b$, kde $\frac{1}{2}b\in\mathbb{N}_0$ a $\frac{1}{2}b<2^{a-1}$. Můžeme použít indukční předpoklad, $J(2^{a-1}+\frac{1}{2}b)=2\cdot\frac{1}{2}b+1=b+1$ a tedy

$$J(n)=J(2m)=2J(m)-1=2(b+1)-1=2b+1,$$

souhlasí.

Jestliže je $n$ liché, tedy $n=2m+1=2^a+b$, pak musí být $b$ liché a $m=2^{a-1}+\frac{1}{2}(b-1)$, kde $\frac{1}{2}(b-1)\in\mathbb{N}_0$ a $\frac{1}{2}(b-1)<2^{a-1}$. Můžeme použít indukční předpoklad, $J(2^{a-1}+\frac{1}{2}(b-1))=2\cdot\frac{1}{2}(b-1)+1=b$ a tedy

$$J(n)=J(2m+1)=2J(m)+1=2b+1,$$

souhlasí. Náš vzorec je tedy správně.

$\triangle$

Tím končí kapitola o lineárních rekurentních rovnicích. Nakonec ještě jedna nepoviná poznámka.

Týká se zásadního problému, že naše krásné metody jsou k ničemu, pokud pravá strana není kvazipolynom, a dokonce ani homogenní rovnici neumíme vyřešit, pokud nejsou koeficienty rovnice konstantní. Touto problematikou se zabývají pokročilé teorie, my si zde ukážeme dva případy, kdy se dají pomocí chytrého převodu použít naše metody.

a) Pokud se v rekurentním vztahu jen násobí/dělí, pak se dá rovnice převést na lineární pomocí logaritmu. Ukážeme to na příkladě.

Máme rovnici $a_{n+1}=\dfrac{a_n^3}{a_{n-1}^2}$. Po jejím zlogaritmování dostáváme $\ln(a_{n+1})=3\ln(a_n)-2\ln(a_{n-1})$. Když si označíme $b_n=\ln(a_n)$, dostáváme tak rovnici $b_{n+1}=3b_n-2b_{n-1}$ neboli $b_{n+2}-3b_{n+1}+2b_n=0$. Charakteristická rovnice $\lambda^2-3\lambda+2=0$ dává $\lambda=1,2$ a řešení $b_n=u1^n+v2^n$. Pak $a_n=e^{b_n}=e^{u+2^n v}=e^u\cdot(e^v)^{2^n}$.

b) Zde se podíváme, co se dá dělat s lineární rekurentní rovnicí 1. řádu, pokud nemá konstantní koeficienty. Uvažujme následující rekurentní vztah: $f(n)a_{n+1}+g(n)a_n=h(n)$.

Vytvoříme funkce $Q(n)=\dfrac{f(1)f(2)\cdots f(n-1)}{g(1)g(2)\cdots g(n-1)g(n)}$, všimněte si, že $Q(n+1)=Q(n)\dfrac{f(n)}{g(n+1)}$. Když použijeme substituci $b_n=g(n)Q(n)a_n$ neboli dosazujeme do rovnice $a_n=\dfrac{b_n}{g(n)Q(n)}$, tak máme

$$f(n)\frac{b_{n+1}}{g(n+1)Q(n+1)}+g(n)\frac{b_n}{g(n)Q(n)}=h(n)\implies b_{n+1}\frac{f(n)g(n+1)}{g(n+1)Q(n)f(n)}+\frac{b_n}{Q(n)}=h(n)$$
$$\implies b_{n+1}+b_n=Q(n)h(n).$$

Dostáváme tedy lineární rekurentní rovnici s konstantními koeficienty, kterou při troše štěstí (když vyjde pravá strana kvazipolynomiální) dokážeme vyřešit. Dokonce lze napsat obecný vzorec. Jestliže začneme počáteční hodnotou $b_1=-C$, pak $b_2=Q(1)h(1)+C$, $b_3=Q(2)h(2)-Q(1)h(1)-C$ atd., indukcí ukážeme vztah $b_n=(-1)^nC-\sum\limits_{i=1}^{n-1}(-1)^{n+i}Q(i)h(i)$. Substituční rovnice pak dává $a_n=\dfrac{(-1)^nC-\sum\limits_{i=1}^{n-1}(-1)^{n+i}Q(i)h(i)}{g(n)Q(n)}$.

Bohužel, míra nutného štěstí je docela vysoká, například rovnice $a_{n+1}-(n+1)a_n=n+1$, kterou jsme dostali v příkladě 10b.g, dává $f(n)=1$, $g(n)=-(n+1)$, $h(n)=n+1$, odtud $Q(n)=\frac{(-1)^n}{(n+1)!}$ a dostáváme rovnici $b_{n+1}+b_n=\frac{(-1)^n}{n!}$. Tím jsme skončili, kvazipolynom na pravé straně není a obecný součtový vzorec výše nám také k přesnému řešení v uzavřeném tvaru nepomůže.

## Cvičení

**Cvičení 10b.1** (rutinní)**:** Každé prázdné pole tabulky reprezentuje rekurentní rovnici, jejíž levou stranu najdete v záhlaví sloupce a pravou stranu napravo v označení řádku. Například první pole druhého (prázdného) řádku dává rovnici $a_{n+2}-4a_{n+1}+3a_n=(-1)^n$.

Pro každé pole najděte odhad tvaru partikulárního řešení (obecný, s konstantami $A, B, \dots$, nemusíte to dál řešit).

| $a_{n+2} - 4a_{n+1} + 3a_n =$ $[\lambda = 1, 3]$ | $a_{n+2} - 2a_n =$ $[\lambda = -2, 2]$ | $a_{n+2} + 2a_{n+1} + a_n =$ $[\lambda = -1\ (2\times)]$ | $L = \big/ = b_n$ |
|---|---|---|---|
| | | | $= n\,2^n$ $[\lambda = 2]$ |
| | | | $= (-1)^n$ $[\lambda = -1]$ |
| | | | $= (n-2)\,3^n$ $[\lambda = 3]$ |
| | | | $= n^2 - 1$ $[\lambda = 1]$ |
| | | | $= 2^n - 2n \cdot (-1)^n$ $[\lambda = 2, 4]$ |
| | | | $= 1 - (-2)^n$ $[\lambda = 1, -2]$ |

**Cvičení 10b.2** (rutinní, zkouškové)**:** Najděte obecná řešení následujících rovnic:
(i) $a_{n+2} - 6a_{n+1} + 8a_n = 0$ pro $n \geq 0$;
(ii) $a_{n+2} - 4a_n = 0$ pro $n \geq 1$;
(iii) $a_{n+2} + a_{n+1} - 2a_n = 0$ pro $n \geq 2$;
(iv) $a_{n+2} - 6a_{n+1} + 9a_n = 0$ pro $n \geq -2$;
(v) $a_{n+2} + 9a_n = 0$ pro $n \geq 0$;
(vi) $a_{n+3} - 2a_{n+2} - a_{n+1} + 2a_n = 0$ pro $n \geq 1$;
(vii) $a_{n+2} - a_n = 18n\,2^n$ pro $n \geq 0$;
(viii) $a_{n+2} + 2a_{n+1} - 3a_n = (5n+12)2^n$ pro $n \geq 0$;
(ix) $a_{n+2} - 4a_{n+1} + 4a_n = 13 \cdot 3^n - 3$ pro $n \geq 1$;
(x) $a_{n+1} = a_n + 2a_{n-1} + 3 \cdot 2^n - 2 \cdot (-2)^n$ pro $n \geq 2$;
(xi) $a_{n+3} + 3a_{n+2} - 4a_n = 16 \cdot 2^n + 9$ pro $n \geq 0$;
(xii) $a_{n+1} = a_{n-1} + n - 1$ pro $n \geq 2$.

**Cvičení 10b.3** (rutinní, zkouškové)**:** Najděte řešení následujících úloh s počátečními podmínkami:
(i) $a_{n+1} = 6a_{n-1} - a_n$ pro $n \geq 3$, $a_2 = -6$, $a_3 = 78$;
(ii) $a_{n+2} - a_n = (8n+18)3^n$ pro $n \geq 1$, $a_1 = 16$, $a_2 = 31$;
(iii) $a_{n+1} = 4a_n - 5a_{n-1} + 2a_{n-2}$ pro $n \geq 2$, $a_0 = 1$, $a_1 = 1$, $a_2 = 2$;
(iv) $a_{n+2} + 4a_n = 0$ pro $n \geq 0$, $a_0 = 0$, $a_1 = 1$;
(v) $a_{n+1} = 2a_n - a_{n-1} - 4 \cdot (-1)^n$ pro $n \geq 0$, $a_{-1} = 0$, $a_0 = 2$;
(vi) $a_{n+2} + 2a_{n+1} - 3a_n = 5 \cdot 2^n + 8$ pro $n \geq 1$, $a_1 = 5$, $a_2 = 9$;
(vii) $a_{n+1} = a_n + 4a_{n-1} - 4a_{n-2} + (6n-7)(-1)^n$ pro $n \geq 2$, $a_0 = 2$, $a_1 = 4$, $a_2 = 7$.

**Cvičení 10b.4** (rutinní, dobré, zkouškové)**:** Uvažujte funkce dané následujícími rekurentními rovnicemi. Pro každou z nich určete asympotickou rychlost růstu (viz značení $\Theta$ z kapitoly 9b) pomocí metod z této kapitoly, ale bez toho, abyste funkce opravdu počítali.
(i) $f(n+1) = 2f(n) + 2n$;
(ii) $f(n+1) = f(n) + 2n$;
(iii) $f(n+1) = 4f(n) - 3f(n-1) + 2n$;
(iv) $f(n+1) = 4f(n) - 3f(n-1) + n2^n$;
(v) $f(n+1) = 4f(n) - 3f(n-1) + 3^n$;
(vi) $f(n+1) = 4f(n) - 3f(n-1) + 4^n$.

**Cvičení 10b.5** (rutinní, dobré, zkouškové)**:** Najděte pomocí rekurentních vztahů explicitní vzorce v uzavřeném tvaru pro následující sumy:
(i) $1 + 4 + 7 + 10 + \dots + (3n+1) = \sum\limits_{k=0}^{n} (3k+1)$;
(ii) $1 + \lambda + \lambda^2 + \dots + \lambda^n = \sum\limits_{k=0}^{n} \lambda^k$ pro $\lambda \neq 1$;
(iii) $1 + 3 + 6 + 10 + \dots + \frac{n(n+1)}{2} = \sum\limits_{k=1}^{n} \frac{k(k+1)}{2}$;
(iv) $1^2 + 3^2 + \dots + (2n+1)^2 = \sum\limits_{k=0}^{n} (2k+1)^2$.

**Cvičení 10b.6** (poučné)**:** Nástupní plat je 20,000. Ve smlouvě je každoroční zvýšení platu o 5 procent kvůli inflaci plus zvýšení o věrnostní částku 1000. Kolik je plat po $n$ letech?

**Cvičení 10b.7** (poučné)**:** Vezmeme si hypotéku $H$ korun. Je úročena měsíčně úrokem $r$ procent, na konci měsíce se platí splátka $S$. Nechť $H(k)$ je dlužná částka po $k$ měsících splácení.
(i) Najděte rekurentní vztah pro $H(k)$, určete počáteční podmínku a vzniklou úlohu vyřešte.
(ii) Určete dlužnou částku po 2 letech, jsou-li údaje $H = 3 \cdot 10^6$, $r = 0.5$ (tedy půl procenta za měsíc), $S = 13000$. Kolik je roční úroková míra?
(iii) Obecně najděte vzorec, jak vysoká musí být splátka $S$, aby dlužná částka ubývala.

**Cvičení 10b.8** (dobré, poučné)**:** Nechť $A_n$ je matice daná $a_{ii} = 2$ pro všechna $i$, $a_{ij} = 1$ pro $|i - j| = 1$ a $a_{ij} = 0$ jinak (tedy 2 na diagonále, 1 v místech hned vedle diagonály a 0 jinde). Najděte rekurentní vzorec pro $d_n = \det(A_n)$.

**Cvičení 10b.9** (rutinní, poučný)**:** Dokažte, že když posloupnost $\{a_n\}_{n=n_0}^{\infty}$ řeší rovnici $a_{n+k}+\sum\limits_{i=0}^{k-1}c_i a_{n+i}=b_n$ a posloupnost $\{\tilde{a}_n\}_{n=n_0}^{\infty}$ řeší rovnici $a_{n+k}+\sum\limits_{i=0}^{k-1}c_i a_{n+i}=\tilde{b}_n$ (stejná levá strana), pak posloupnost $\{a_n+\tilde{a}_n\}_{n=n_0}^{\infty}$ řeší rovnici $a_{n+k}+\sum\limits_{i=0}^{k-1}c_i a_{n+i}=b_n+\tilde{b}_n$.

**Řešení:**

**10b.1:**

| $a_{n+2}-4a_{n+1}+3a_n=$ $[\lambda=1,3]$ | $a_{n+2}-2a_n=$ $[\lambda=-2,2]$ | $a_{n+2}+2a_{n+1}+a_n=$ $[\lambda=-1\ (2\times)]$ | $L= / =b_n$ |
|---|---|---|---|
| $(An+B)2^n$ | $n(An+B)2^n$ | $n^2(An+B)2^n$ | $=n\,2^n$ $[\lambda=2]$ |
| $A(-1)^n$ | $A(-1)^n$ | $n^2A(-1)^n$ | $=(-1)^n$ $[\lambda=-1]$ |
| $n(An+B)3^n$ | $(An+B)3^n$ | $n^2(An+B)3^n$ | $=(n-2)\,3^n$ $[\lambda=3]$ |
| $n(An^2+Bn+C)$ | $(An^2+Bn+C)$ | $(An^2+Bn+C)$ | $=n^2-1$ $[\lambda=1]$ |
| $A2^n+(Bn+C)(-1)^n$ | $A2^n+n(Bn+C)(-1)^n$ | $A2^n+n^2(Bn+C)(-1)^n$ | $=2^n-2n\cdot(-1)^n$ $[\lambda=2,4]$ |
| $nA+B(-2)^n$ | $A+nB(-2)^n$ | $A+B(-2)^n$ | $=1-(-2)^n$ $[\lambda=1,-2]$ |

**10b.2:** (i): $(\lambda-2)(\lambda-4)=0$, $\{2^nu+4^nv\}_{n=0}^{\infty}$; (ii): $(\lambda-2)(\lambda+2)=0$, $\{2^nu+(-2)^nv\}_{n=1}^{\infty}$;
(iii): $(\lambda+2)(\lambda-1)=0$, $\{u+(-2)^nv\}_{n=2}^{\infty}$; (iv): $(\lambda-3)^2=0$, $\{(un+v)3^n\}_{n=-2}^{\infty}$;
(v): $(\lambda-3i)(\lambda+3i)=0$, $\big\{\big(u\cos\big(n\frac{\pi}{2}\big)+v\sin\big(n\frac{\pi}{2}\big)\big)3^n\big\}_{n=0}^{\infty}$;
(vi): $(\lambda-1)(\lambda+1)(\lambda-2)=0$, $\{u+(-1)^nv+2^nw\}_{n=1}^{\infty}$;
(vii): $(\lambda-1)(\lambda+1)=0$, $a_{h,n}=u+(-1)^nv$; odhad $a_n=(An+B)2^n$, $\{(6n-16)2^n+u+(-1)^nv\}_{n=0}^{\infty}$;
(viii): $(\lambda-1)(\lambda+3)=0$, $a_{h,n}=u+(-3)^nv$; odhad $a_n=(An+B)2^n$, $\{n2^n+u+(-3)^nv\}_{n=0}^{\infty}$;
(ix): $(\lambda-2)^2=0$, $a_{h,n}=n2^nu+2^nv$; odhad $a_n=A3^n+B$, $\{13\cdot 3^n-3+n2^nu+2^nv\}_{n=1}^{\infty}$;
(x): přepis: $a_{n+2}-a_{n+1}-2a_n=6\cdot 2^n+4\cdot(-2)^n$ pro $n\geq 1$; $(\lambda-2)(\lambda+1)=0$, $a_{h,n}=2^nu+(-1)^nv$; odhad $a_n=n^1A2^n+B(-2)^n$, $\{n2^n+(-2)^n+2^nu+(-1)^nv\}_{n=1}^{\infty}$;
(xi): $(\lambda-1)(\lambda+2)^2=0$, $a_{h,n}=u+(-2)^nv+n(-2)^nw$; odhad $a_n=A2^n+n^1B$, $\{n+2^n+u+(-2)^nv+n(-2)^nw\}_{n=0}^{\infty}$;
(xii): přepis: $a_{n+2}-a_n=n=n\cdot 1^n$ pro $n\geq 1$; $(\lambda-1)(\lambda+1)=0$, $a_{h,n}=u+(-1)^nv$; odhad $a_n=n(An+B)$, $\big\{\frac{1}{4}n^2-frac12n+u+(-1)^nv\big\}_{n=0}^{\infty}$.
**10b.3:** (i): přepis: $a_{n+2}+a_{n+1}-6a_n=0$ pro $n\geq 2$; $(\lambda-2)(\lambda+3)=0$, $\{2^nu+(-3)^nv\}_{n=2}^{\infty}$, poč. podm. dávají $\{3\cdot 2^n-2(-3)^n\}_{n=2}^{\infty}$;
(ii): $(\lambda-1)(\lambda+1)=0$, $a_{h,n}=u+(-1)^nv$; odhad $a_n=(An+B)3^n$, $\{n3^n+u+(-1)^nv\}_{n=1}^{\infty}$, poč. podm. dávají $\{n3^n+13\}_{n=1}^{\infty}$;
(iii): přepis: $a_{n+3}-4a_{n+2}+5a_{n+1}-2a_n=0$ pro $n\geq 0$; $(\lambda-1)^2(\lambda-2)=0$; $\{u+nv+2^nw\}_{n=0}^{\infty}$; poč. podm. dávají $\{2^n-n\}_{n=0}^{\infty}$;
(iv): $(\lambda-2i)(\lambda+2i)=0$, $\big\{\big(u\cos\big(n\frac{\pi}{2}\big)+v\sin\big(n\frac{\pi}{2}\big)\big)2^n\big\}_{n=0}^{\infty}$; poč. podm. dávají $\big\{2^{n-1}\sin\big(n\frac{\pi}{2}\big)\big\}_{n=0}^{\infty}$, je to $\{0,1,0,-4,0,16,0,-64,0,\dots\}$;
(v): přepis: $a_{n+2}-2a_{n+1}+a_n=4\cdot(-1)^n$ pro $n\geq -1$; $(\lambda-1)^2=0$, $a_{h,n}=u+nv$; odhad $a_n=A(-1)^n$, $\{(-1)^n+u+nv\}_{n=-1}^{\infty}$, poč. podm. dávají $\{(-1)^n+1\}_{n=-1}^{\infty}$, je to $\{0,2,0,2,0,2,0,2,\dots\}$;
(vi): $(\lambda-1)(\lambda+3)=0$, $a_{h,n}=u+(-3)^nv$; odhad $a_n=A2^n+n^1B$, $\{2^n+2n+u+(-3)^nv\}_{n=1}^{\infty}$, poč. podm. dávají $\{2^n+2n+1\}_{n=1}^{\infty}$;
(vii): přepis: $a_{n+3}-a_{n+2}-4a_{n+1}+4a_n=(6n+5)(-1)^n$ pro $n\geq 0$; $(\lambda-1)(\lambda-2)(\lambda+2)=0$, $a_{h,n}=u+2^nv+(-2)^nw$; odhad $a_n=(An+B)(-1)^n$, $\{(n+1)(-1)^n+u+2^nv+(-2)^nw\}_{n=0}^{\infty}$, poč. podm. dávají $\{(n+1)(-1)^n+2^{n+1}-(-2)^n\}_{n=0}^{\infty}$.
**10b.4:** (i): char. čísla: $\lambda=2$, proto hom. řeš. $f(n)=2^nu$. Odhad pravé strany: $f(n)=An+B$, proto obecné řešení bude tvaru $f(n)=2^n+An+B$. Protože $2^n\gg(2n+1)$, bude $f(n)=\Theta(2^n)$.
(ii): char. čísla: $\lambda=1$, proto hom. řeš. $f(n)=u$. Odhad pravé strany: $f(n)=n(An+B)$, proto obecné řešení bude tvaru $f(n)=u+An^2+Bn$. Máme $f(n)=\Theta(n^2)$.
(iii): char. čísla: $\lambda=1,3$, proto hom. řeš. $f(n)=u+3^nv$. Odhad pravé strany: $f(n)=n(An+B)$, proto obecné řešení bude tvaru $f(n)=u+3^nv+An^2+Bn$. Protože $3^n\gg(An^2+Bn+u)$, bude $f(n)=\Theta(3^n)$.

(iv): char. čísla: $\lambda = 1, 3$, proto hom. řeš. $f(n) = u + 3^n v$. Odhad pravé strany: $f(n) = (An + B)2^n$, proto obecné řešení bude tvaru $f(n) = u + 3^n v + (An + B)2^n$. Protože $3^n \gg (An2^n + B2^n + u)$, bude $f(n) = \Theta(3^n)$.
(v): char. čísla: $\lambda = 1, 3$, proto hom. řeš. $f(n) = u + 3^n v$. Odhad pravé strany: $f(n) = An\,3^n$, proto obecné řešení bude tvaru $f(n) = u + 3^n v + An\,3^n$. Máme $f(n) = \Theta(n\,3^n)$.
(vi): char. čísla: $\lambda = 1, 3$, proto hom. řeš. $f(n) = u + 3^n v$. Odhad pravé strany: $f(n) = A \cdot 4^n$, proto obecné řešení bude tvaru $f(n) = u + 3^n v + A \cdot 4^n$. Protože $4^n \gg (3^n + u)$, bude $f(n) = \Theta(4^n)$.
**10b.5:** (i): $s_{n+1} = s_n + (3n + 4)$ a $s_0 = 1$; $s_{h,n} = u$, odhad $s_n = n(An + B) = An^2 + Bn$, po dosazení $A = \frac{3}{2}$, $B = \frac{5}{2}$, $s_n = \frac{3}{2}n^2 + \frac{5}{2}n + u$. Poč. podm. dá $u = 1$, $s_n = \frac{(n+1)(3n+2)}{2}$.
(ii): $s_{n+1} = s_n + \lambda^{n+1}$ a $s_0 = 1$; $s_{h,n} = u$, odhad pro $s_{n+1} - s_n = \lambda \cdot \lambda^n$ a $\lambda \neq 1$ je $s_n = A\lambda^k$, po dosazení $A = \frac{\lambda}{\lambda-1}$, $s_n = \frac{\lambda}{\lambda-1}\lambda^n + u$. Poč. podm. dá $u = -\frac{1}{\lambda-1}$, $s_n = \frac{1-\lambda^{n+1}}{1-\lambda}$.
(iii): $s_{n+1} = s_n + \frac{(n+1)(n+2)}{2}$ a $s_1 = 1$; $s_{h,n} = u$, odhad $s_n = n(An^2 + Bn + C) = An^3 + Bn^2 + Cn$, po dosazení $A = \frac{1}{6}$, $B = \frac{1}{2}$, $C = \frac{1}{3}$, $s_n = \frac{1}{6}n^3 + \frac{1}{2}n^2 + \frac{1}{3}n + u$. Poč. podm. dá $u = 0$, $s_n = \frac{n(n+1)(n+2)}{6}$.
(iv): $s_{n+1} = s_n + (2n + 3)^2$ a $s_0 = 1$; $s_{h,n} = u$, odhad $s_n = n(An^2 + Bn + C) = An^3 + Bn^2 + Cn$, po dosazení $A = \frac{4}{3}$, $B = 4$, $C = \frac{11}{3}$, $s_n = \frac{4}{3}n^3 + 4n^2 + \frac{11}{3}n + u$. Poč. podm. dá $u = 1$, $s_n = \frac{(n+1)(2n+1)(2n+3)}{3}$.
**10b.6:** $P(n + 1) = 1.05 \cdot P(n) + 1000$, $P(0) = 20000$.
Homogenní: $P(n + 1) - 1.05 \cdot P(n) = 0$, $\lambda = 1.05$, $P_h(n) = (1.05)^n$.
Odhad kvazipolynomiální pravé strany: $P(n) = A$, dosadit, $-0.05A = 1000$, $A = -20000$, proto $P(n) = u(1.05)^n - 20000$. Poč. podmínka dá $40000 = u$. Proto $P(n) = 40000 \cdot (1.05)^n - 20000$.
**10b.7:** (i): $H(k + 1) = \bigl(1 + \frac{r}{100}\bigr)H(k) - S$, $H(0) = H$; označme $R = 1 + \frac{r}{100}$.
Homogenní: $H(k + 1) - RH(k) = 0$, $\lambda = R$, $H_h(k) = uR^k$. Toto popisuje růst dluhu bez splátek.
Odhad kvazipolynomiální pravé strany: $H(k) = A$, dosadit, $(1 - R)A = -S$, $A = \frac{S}{R-1}\frac{100S}{r}$, $H(k) = uR^k + \frac{100S}{r}$.
Poč. podmínka: $H - \frac{100S}{r} = u$. Proto $H(k) = \bigl(H - \frac{100S}{r}\bigr)\bigl(1 + \frac{r}{100}\bigr)^k + \frac{100S}{r}$.
(ii): $H(k + 1) - 1.005H(k) = 15000$, $H(0) = 3000000$;
Kdyby se nic nesplácelo, byla by dlužná částka za rok $(1.005)^{12}H(0)$, tedy roční úrok je $100\cdot((1.005)^{12} - 1) \sim 6.17$.
Po dvou letech je dluh $H(24) = 400000 \cdot (1.005)^{24} + 2600000 \sim 3025000$.
(iii): Je potřeba $H - \frac{100S}{r} < 0$ tedy $S > \frac{rH}{100}$.
**10b.8:** Rozvoj podle prvního sloupce a pak prvního řádku

$$d_{n+1} = \begin{vmatrix} 2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 2 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 \end{vmatrix} = 2 \cdot \begin{vmatrix} 2 & 1 & 0 & \cdots & 0 \\ 1 & 2 & 1 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{vmatrix} - 1 \cdot \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 2 & 1 & \cdots & 0 \\ 0 & 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{vmatrix} = 2d_n - 1 \cdot 1 \cdot \begin{vmatrix} 2 & 1 & \cdots & 0 \\ 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \cdots & 2 \end{vmatrix},$$

tedy $d_{n+1} = 2d_n - d_{n-1}$.
**10b.9:** Dosaďte $a_n + \tilde{a}_n$ do levé strany rovnice, roznásobte, shromážděte k sobě všechna $a_n$ a všechna $\tilde{a}_n$ a pak použijte předpoklad, viz důkaz Věty 10a.4.

## 10c. Další rovnice (Master theorem)

Mnohé rekurzivní algoritmy používají metodu „rozděl a panuj“ (divide-and-conquer). Problém velikosti $n$ se rozdělí na $a$ menších problémů velikosti $\frac{n}{b}$ (reálně nejbližší celé číslo k $\frac{n}{b}$, třeba $\lceil \frac{n}{b} \rceil$). Ty se vyřeší a jednotlivé výsledky je pak ještě třeba zpracovat, to taky něco stojí, takže typická rovnice pro náročnost je $a_n = a \cdot a_{n/b} + g(n)$. Bývá tradiční používat v této souvislosti funkce, takže spíš vidíme rovnici $f(n) = a \cdot f\bigl(\frac{n}{b}\bigr) + g(n)$. Tato rozhodně není lineární.

**!** **Příklad 10c.a:** Zde se podíváme, kolik porovnávání nás stojí binární vyhledávání v seznamu o délce $n$, označme to $f(n)$. Binární vyhledávání pracuje s uspořádaným seznamem (dle abecedy, velikosti atd.), takže když hledáme v seznamu o velikosti $n$, můžeme se podívat na prvek uprostřed a hned zjistíme, zda hledaný objekt je v první nebo druhé polovině seznamu. Na příslušnou polovinu (o velikosti $\frac{n}{2}$) pak zase pošleme binární vyhledávání, pokud ale není prázdná (pak bychom řekli, že hledaný objekt v seznamu není, ověření nás stojí další porovnání). Vidíme, že hledání v seznamu o $n$ položkách vyžaduje tolik porovnávání, kolik hledání v seznamu polovičním plus dvě porovnání, máme tedy $f(n) = f\bigl(\frac{n}{2}\bigr) + 2$, což je přesně rovnice typu, který v této kapitole budeme zkoumat. Protože jsme o tom ještě nic nevymysleli, zkusíme na to jít selským rozumem.

Nejprve se zbavíme zlomku. Problém, zda zaokrouhlovat nahoru či dolů, vyřešíme elegantně tím, že se omezíme (zatím) na $n$ sudá, pak lze rovnici přepsat jako $f(2n) = f(n) + 2$. Kdybychom použili posloupnosti, dostali bychom

$a_{2n} = a_n + 2$, což je opravdu rekurentní rovnice, ale rozhodně není lineární. Vidíme, že problémy tohoto typu jsou něčím jiným, než jsme zatím probírali.

Výhoda nového zápisu $f(2n) = f(n) + 2$ je v tom, že jsme se vyhli zlomkům. Nevýhoda je, že nám tato rovnice neumožňuje zjistit hodnoty pro všechna $n$. Zkusme se intuitivně podívat, co pro neznámou funkci dostáváme.

Začneme počáteční hodnotou pro jednoprvkový seznam $f(1) = 1$ (prostě se podíváme, jestli ten jeden prvek v seznamu je nebo není to, co hledáme). Pak umíme najít $f(2) = f\big(\frac{2}{2}\big) + 2 = f(1) + 2 = 1 + 2$, podobně $f(4) = f\big(\frac{4}{2}\big) + 2 = f(2) + 2 = (1 + 2) + 2$, $f(8) = f\big(\frac{8}{2}\big) + 2 = (1 + 2 + 2) + 2$, $f(16) = (1 + 2 + 2 + 2) + 2$ atd. Naopak třeba $f(6)$ nezískáme, protože $f(6) = f(3) + 2$ a $f(3)$ neznáme. Vidíme, že dotyčný rekurentní vztah nám dává $f(n)$ pro $n$ typu $n = 2^k$, můžeme si tipnout, že pak $f(n) = 2k + 1$. Tento výsledek není uspokojivý, protože se v něm znenadání objevuje $k$, zatímco proměnná ve funkci je $n$. Z rovnice $n = 2^k$ si ovšem $k$ dokážeme vyjádřit jako $k = \log_2(n)$ a dostaneme $f(n) = 2\log_2(n) + 1$. To znamená, že binární vyhledávání je dost rychlé, podstatně rychlejší než přímá úměra k délce seznamu.

Jaká je náročnost pro jiná $n$? To je něco, co nám dotyčná rovnice přímo neřekne, záleží to na konkrétní interpretaci toho zlomku $\frac{n}{b}$. Zatím to odložíme.

Shrňme si poznatky, které budou hrát významnou roli ve zbytku kapitoly: Jde o zcela nový typ problému, který po vhodném přepisu do tvaru $f(bn) = af(n) + g(bn)$ dává (při troše štěstí) rozumným způsobem hodnoty pro proměnné $n = b^k$. Určení $f(n)$ pro ostatní $n$ není zjevné a později na tom budeme muset zapracovat.
△

Rovnice tohoto typu jsou v teorii algoritmů velice užitečné (viz další příklady či cvičení 10c.4). Než začneme vytvářet teorii, podíváme se na „homogenní rovnici“, tedy na případ, kdy $g(n) = 0$. Zkušenost říká, že by měl být nejsnažší.

**!** **Příklad 10c.b:** Předpokládejme, že funkce $f$ je daná vztahem $f(n) = a \cdot f\big(\frac{n}{b}\big)$, kde $a > 0$ a $b \in \mathbb{N}$, $b \geq 2$. Zkusíme nalézt $f$ pomocí iterování vztahu z definice.

Jestliže je číslo $n$ ve tvaru $n = b^k$, pak můžeme rekurzí najít

$$f(b) = af\Big(\frac{b}{b}\Big) = af(1),\ f(b^2) = af\Big(\frac{b^2}{b}\Big) = af(b) = a^2 f(1),\ f(b^3) = af(b^2) = a^3 f(1),$$
$$f(b^k) = af(b^{k-1}) = a^2 f(b^{k-2}) = \cdots = a^{k-1} f(b) = a^k f(1).$$

Důkaz správnosti vzorce provedeme indukcí na $k$, dokazujeme tvrzení $V(k)$: $f(b^k) = a^k f(1)$.

(0) Pro $k = 0$ platí $f(b^0) = f(1) = a^0 f(1)$, to souhlasí.

(1) Předpokládejme, že pro nějaké $k \in \mathbb{N}_0$ platí $f(b^k) = a^k f(1)$. Pak podle definice

$$f(b^{k+1}) = a \cdot f\big(\tfrac{b^{k+1}}{b}\big) = a \cdot f(b^k) = a \cdot a^k f(1) = a^{k+1} f(1).$$

Dokázali jsme tedy pro všechna $k \in \mathbb{N}_0$ implikaci $V(k) \implies V(k+1)$, čímž je důkaz indukcí ukončen a výsledek $f(n) = a^k f(1)$ je potvrzen. Opět se zbavíme $k$ ve výsledku: Jestliže $n = b^k$, pak $k = \log_b(n)$, po dosazení máme $f(n) = a^{\log_b(n)} f(1)$.

Kam tato funkce zapadá na naší obvyklé škále (mocniny, geometrické posloupnosti, faktoriály atd.)? To uvidíme, když si tento výraz přepíšeme:

$$a^{\log_b(n)} = \big(b^{\log_b(a)}\big)^{\log_b(n)} = b^{\log_b(a)\log_b(n)} = \big(b^{\log_b(n)}\big)^{\log_b(a)} = n^{\log_b(a)}.$$

Proto pro čísla typu $n = b^k$ platí $f(n) = n^{\log_b(a)} f(1)$. Toto je velice výhodný tvar, čísla $a, b$ jsou totiž konstanty známé ze zadání, takže $n^{\log_b(a)} f(1)$ je jistá mocnina $n$.

Dá se ukázat (uděláme to pro obecnější případy níže), že se tato rychlost růstu zachová i pro ostatní čísla $n \in \mathbb{N}$. Funkce dané touto nejjednodušší rovnicí tedy rostou polynomiální rychlostí $n^{\log_b(a)}$ neboli $f(n) = \Theta(n^{\log_b(a)})$ (viz kapitola 9b).

Je dobré si všimnout, že pokud $a = 1$, tak dostáváme $\Theta(n^0) = \Theta(1)$, což říká, že funkce by měla být omezená a nikam nerůst. Když použijeme přesné vyjádření výsledku, tak dokonce vidíme, že $f$ je přesně konstantní funkce na $M$: $f(n) = a^k f(1) = f(1)$. Je to ostatně vidět i přímo z rovnice, která pak říká $f(bn) = f(n)$ a tedy $f(b) = f(1)$, $f(b^2) = f(b) = f(1)$ atd.
△

Tento příklad byl důležitý, protože ukázal hlavní ingredience naší další práce, zejména ten závěrečný převod na mocninu $n$ budeme opakovaně používat. Raději si jej zvýrazníme:

**Fakt 10c.1.**
Nechť $b \in \mathbb{N}$ splňuje $b \geq 2$, nechť $a > 0$. Pro všechna $n \in \mathbb{N}$ platí: Jestliže $n = b^k$ pro nějaké $k \in \mathbb{N}_0$, pak $a^k = n^{\log_b(a)}$.

Dalším a hlavním úkolem je zjistit, co se stane, když $g(n)$ není nulové. Postup z příkladu 10c.b lze zopakovat a dostaneme obecný výsledek.

**Lemma 10c.2.**
Nechť $b \in \mathbb{N}$, $b \geq 2$. Nechť $f(n)$ je funkce definovaná pro $n \in M = \{b^k;\ k \in \mathbb{N}_0\}$. Předpokládejme, že existuje $a \in \mathbb{R}$, $a > 0$ a funkce $g$ na $M$ taková, že

$$f(n) = a \cdot f\Bigl(\frac{n}{b}\Bigr) + g(n) \qquad \text{pro všechna } n \in M,\ n \geq 1.$$

Pak pro $n = b^k \in M$ platí

$$f(n) = a^k f(1) + \sum_{i=0}^{k-1} a^i g\Bigl(\frac{n}{b^i}\Bigr).$$

**Důkaz** (poučný)**:** Zase aplikujeme postup s opakovaným použitím rekurentní rovnice, dokud nedojdeme k $f(1)$.

$$\begin{aligned} f(n) &= af\Bigl(\frac{n}{b}\Bigr) + g(n) = a\Bigl[af\Bigl(\frac{\frac{n}{b}}{b}\Bigr) + g\Bigl(\frac{n}{b}\Bigr)\Bigr] + g(n) = a^2 f\Bigl(\frac{n}{b^2}\Bigr) + ag\Bigl(\frac{n}{b}\Bigr) + g(n) \\ &= a^2\Bigl[af\Bigl(\frac{\frac{n}{b^2}}{b}\Bigr) + g\Bigl(\frac{n}{b^2}\Bigr)\Bigr] + ag\Bigl(\frac{n}{b}\Bigr) + g(n) = a^3 f\Bigl(\frac{n}{b^3}\Bigr) + a^2 g\Bigl(\frac{n}{b^2}\Bigr) + ag\Bigl(\frac{n}{b}\Bigr) + g(n) = \dots \\ &= a^k f\Bigl(\frac{n}{b^k}\Bigr) + a^{k-1} g\Bigl(\frac{n}{b^{k-1}}\Bigr) + \dots + a^2 g\Bigl(\frac{n}{b^2}\Bigr) + ag\Bigl(\frac{n}{b}\Bigr) + a^0 g\Bigl(\frac{n}{b^0}\Bigr) \quad \text{a} \quad \frac{a}{b^k} = 1. \end{aligned}$$

Ta pasáž se třemi tečkami samozřejmě způsobuje, že toto není žádný důkaz. Poskytlo nám to nicméně kandidáta, správnost vzorce teď dokážeme indukcí na $k$. Ukážeme, že funkce $f$ daná tímto vzorcem splňuje danou rovnici.

(0) $k = 0$: $f(b^0) = a^0 f(1) + \sum\limits_{i=0}^{-1} a^i g\Bigl(\frac{b^0}{b^i}\Bigr) = f(1) + 0 = f(1)$.

V té sumě se sčítá přes prázdnou množinu, což je automaticky 0.

(1) Předpokládejme, že platí $f(b^k) = a^k f(1) + \sum\limits_{i=0}^{k-1} a^i g\Bigl(\frac{b^k}{b^i}\Bigr)$. Pak máme

$$\begin{aligned} f(b^{k+1}) &= a \cdot f\Bigl(\frac{b^{k+1}}{b}\Bigr) + g(b^{k+1}) = a \cdot f(b^k) + g(b^{k+1}) = a \cdot \Bigl(a^k f(1) + \sum_{i=0}^{k-1} a^i g\Bigl(\frac{b^k}{b^i}\Bigr)\Bigr) + g(b^{k+1}) \\ &= a^{k+1} f(1) + \sum_{i=0}^{k-1} a^{i+1} g\Bigl(\frac{b^{k+1}}{b^{i+1}}\Bigr) + g(b^{k+1}) = \left| \begin{array}{c} j = i+1 \\ i = 0 \implies j = 1 \\ i = k-1 \implies j = k \end{array} \right| \\ &= a^{k+1} f(1) + \sum_{j=1}^{k} a^j g\Bigl(\frac{b^{k+1}}{b^j}\Bigr) + a^0 g\Bigl(\frac{b^{k+1}}{b^0}\Bigr) = a^{k+1} f(1) + \sum_{j=0}^{(k+1)-1} a^j g\Bigl(\frac{b^{k+1}}{b^j}\Bigr). \end{aligned}$$

Správnost vzorce je dokázána.

□

Tento vzorec nedává hledanou funkci v pěkném tvaru, naštěstí umíme pro nejobvyklejší typy funkce $g(n)$ najít kompaktnější vyjádření.

**! Příklad 10c.c:** Nechť $b \in \mathbb{N}$, $b \geq 2$, uvažujme funkci $f$ danou na množině $M = \{b^k;\ k \in \mathbb{N}_0\}$ rekurzivním vztahem $f(n) = a \cdot f\bigl(\frac{n}{b}\bigr) + g(n)$ pro $n \geq 1$.

**1)** Nejprve analyzujeme situaci, kdy $g(n) = c$ pro nějaké $c \in \mathbb{R}$. Lemma 10c.2 říká, že pro $n = b^k$ máme $f(n) = a^k f(1) + \sum\limits_{i=0}^{k-1} a^i c$. Teď jsou dvě možnosti.

**1a)** Pokud $a = 1$, tak dostaneme $f(n) = f(1) + kc = f(1) + c\log_b(n)$. Funkce tedy na $M$ roste logaritmickou rychlostí.

Poznamenejme, že základ logaritmu zde nehraje zásadní roli, protože jej umíme převést na jakýkoliv jiný díky vzorečku $\log_b(n) = c_{b,B}\log_B(n)$, kde $c_{b,B} = \log_b(B)$. V computer science jsou populární logaritmy o základu 2, takže můžeme psát $f(n) = f(1) + c\log_b(2)\log_2(n)$. Vzhledem k tomu, že při zkoumání růstu funkcí nás násobící kladné konstanty nezajímají, nám ten člen $\log_b(2)$ nevadí. Lze tedy konstatovat, že $f(n) = \Theta(\log_2(n))$ pro $n \in M$.

Mimochodem, přesně sem zapadá rovnice z příkladu 10c.a, výsledky souhlasí.

**1b)** Pokud $a > 1$, pak pomocí vzorce z Věty 9c.2 a Faktu 10c.1 dostáváme

$$f(n) = a^k f(1) + c\frac{1-a^k}{1-a} = a^k\Big(f(1) - \frac{c}{1-a}\Big) + \frac{c}{1-a} = n^{\log_b(a)}\Big(f(1) - \frac{c}{1-a}\Big) + \frac{c}{1-a}.$$

Zatímco tedy v prvním případě $a = 1$ platí $f(n) = \Theta(\log_2(n))$ (logaritmický růst), ve druhém případě máme $f(n) = \Theta(n^{\log_b(a)})$ (polynomiální růst), zatím samozřejmě pouze na $M$.

Všimněte si, že ve výrazu $n^{\log_b(a)}$ už se základem logaritmu pohybovat beztrestně nelze, protože vzniklá konstanta už nebude nezajímavá, ale podstatně ovlivní rychlost růstu tím, že nám umocní $n$: $n^{\log_b(n)} = (n^{\log_b(2)})^{\log_2(n)}$. Takže to necháme pěkně tak, jak to vyšlo.

Ještě bychom měli prozkoumat případ $0 < a < 1$, ale nemá to moc smysl. Pak totiž dostáváme stejný vzorec jako v části 1b) a pro $0 < a < 1$ v něm platí $a^k \to 0$, pro velká $n$ je tedy funkce $f$ v zásadě konstantní. Najdou se algoritmy, které trvají v zásadě stále stejně, ať je na vstupu cokoliv (třeba algoritmus, který na každý vstup reaguje vypsáním „Dnes nemám náladu“ a skončí), ale na takové algoritmy asi přes rekurentní vzorce stejně nepůjdeme. Většina autorů proto předpokládá automaticky, že ve studovaných rovnicích je $a \geq 1$, začneme to také dělat.

Poslední poznámka: Jak to bude s výsledky, když $c = 0$, neboli máme homogenní případ řešený v příkladě 10c.b? Dosazením $c = 0$ do vzorců, které jsme zde odvodili, dostaneme stejné výsledky jako v onom příkladě, takže je vše v pořádku.

**2)** Teď budeme analyzovat situaci, kdy $g(n) = cn^d$ pro nějaké $c \in \mathbb{R} - \{0\}$, $d \in \mathbb{N}$. Lemma 10c.2 pak pro $n = b^k$ dává

$$f(n) = a^k f(1) + \sum_{i=0}^{k-1} a^i c\Big(\frac{n}{b^i}\Big)^d = a^k f(1) + cn^d \sum_{i=0}^{k-1}\Big(\frac{a}{b^d}\Big)^i.$$

Opět sčítáme geometrickou posloupnost, takže musíme rozebrat dva případy.

**2a)** Jestliže $a = b^d$, pak dostaneme

$$f(n) = a^k f(1) + cn^d\sum_{i=0}^{k-1} 1 = a^k f(1) + cn^d k = n^{\log_b(a)} f(1) + cn^d \log_b(n).$$

Všimněte si, že $a = b^d$ znamená $\log_b(a) = d$, takže dostáváme $f(n) = n^d f(1) + cn^d \log_b(n)$.

**2b)** Druhý případ je, že $a \neq b^d$, tedy $\frac{a}{b^d} \neq 1$ a geometrický součet získáme pomocí vzorce z Věty 9c.2. V následných úpravách se nám pak kromě Faktu 10c.1 bude hodit pozorování, že $(b^d)^k = (b^k)^d = n^d$.

$$\begin{aligned} f(n) &= a^k f(1) + cn^d \cdot \frac{1 - (\frac{a}{b^d})^k}{1 - \frac{a}{b^d}} = a^k f(1) + cn^d \cdot \frac{b^d - b^d \frac{a^k}{(b^d)^k}}{b^d - a} = a^k f(1) + cn^d \cdot \frac{b^d - b^d\frac{a^k}{n^d}}{b^d - a} \\ &= a^k f(1) - n^d \cdot \frac{a^k}{n^d}\frac{cb^d}{b^d - a} + n^d \cdot \frac{cb^d}{b^d - a} = n^{\log_b(a)}\Big(f(1) - \frac{cb^d}{b^d - a}\Big) + n^d \cdot \frac{cb^d}{b^d - a}. \end{aligned}$$

**3)** První část byla dobrá jako příprava na těžší část 2), nicméně se nabízí otázka, zda to nebylo zbytečné. Když ve výsledku části 2) použijeme $d = 0$, dostaneme stejné vzorce jako v části 1)? Ano. Je tedy možné udělat obecný závěr.

**Výsledek**: Nechť $g(n) = cn^d$ pro $c \in \mathbb{R} - \{0\}$ a $d \in \mathbb{N}_0$.
Jestliže $a = b^d$, tak $f(n) = n^d f(1) + cn^d \log_b(n)$.
Jestliže $a \neq b^d$, tak $f(n) = n^{\log_b(a)}\Big(f(1) - \frac{cb^d}{b^d - a}\Big) + n^d\frac{cb^d}{b^d - a}$.

Všimněte si, že jsme vyloučili možnost $c = 0$. V této verzi výsledku se totiž soustředíme na parametr $d$, což u homogenní rovnice nemá smysl, třeba proto, že libovolné $d$ bude vyhovovat: Platí $g(n) = 0 = 0n^d$ pro všechna $d$, čímž by vznikl problém se zařazením do správné varianty v našem výsledku. Všimněte si nicméně, že když dosadíme $c = 0$ do druhého vzorce, tak dostaneme přesně výsledek z příkladu 10c.b. Je tedy možné do tohoto případu zahrnout i možnost $c = 0$, bude se nám to hodit později.
△

Příklad nám poskytl zajímavé vzorce, zásadní problém ovšem je, že jsme dostali funkci pouze na množině $M$, ale algoritmy se používají pro $n$ i mimo tuto množinu. Funkce náročnosti tam má určitě nějakou hodnotu, ale z rekurentní rovnice ji nedostaneme, záleží to na tom, jak se v konkrétním algoritmu řeší rozdělování na $b$ částí. Zde si pomůžeme dvěma zjednodušeními. Za prvé, nás vlastně často nezajímá přesný vzorec pro $f$, ale odhad rychlosti růstu, což už jsme ostatně dělali v předchozích příkladech nebo ve cvičení 10b.3. Díky tomu nepotřebujeme úplně přesnou informaci, stačí jen přibližná. Tu získáme z další úvahy, je totiž rozumné předpokládat, že $f$ nikde neklesá (což se u náročnosti algoritmu dá očekávat). Pak už lze výsledky získané v příkladě 10c.c vztáhnout na všechna $n$.

! **Věta 10c.3.** (The Master theorem)
Předpokládejme, že neklesající nezáporná funkce $f$ na $\mathbb{N}$ splňuje rovnici $f(n)=a\cdot f\bigl(\frac{n}{b}\bigr)+cn^d$ na množině $M=\{b^k;\ k\in\mathbb{N}\}$, kde $b\in\mathbb{N}$ splňuje $b\geq 2$ a $a,c\in\mathbb{R}$, $d\in\mathbb{N}_0$ jsou konstanty splňující $a\geq 1$ a $c>0$. Pak platí následující:
(i) Jestliže $a>b^d$, tak $f(n)=\Theta(n^{\log_b(a)})$.
(ii) Jestliže $a=b^d$, tak $f(n)=\Theta(n^d\log_2(n))$.
(iii) Jestliže $a<b^d$, tak $f(n)=\Theta(n^d)$.

**Důkaz** (poučný)**:** (i): Jestliže $a>b^d$, tak podle příkladu 10c.c máme pro $n\in M$ vzorec $f(n)=c_1n^{\log_b(a)}+c_2n^d$, kde $c_1,c_2\in\mathbb{R}$ jsou konstanty nezávisející na $n$. Z $a>b^d$ plyne $\log_b(a)>d$, proto první sčítanec převáží nad druhým a funkce $f(n)$ roste jako $c_1n^{\log_b(a)}$, pokud tedy toto číslo není nula. Co o $c_1$ víme? Máme $c_1=f(1)-\dfrac{cb^d}{b^d-a}$, kde $f(1)\geq 0$ ($f$ je nezáporná), zlomek je pak díky předpokladům $c>0$, $b>0$ a $a>b^d$ záporný a odčítáme jej, tedy $c_1>0$. První člen proto nezmizí a je násoben kladným číslem, lze tedy říct, že funkce $f(n)$ roste jako $n^{\log_b(a)}$.

Proto existují konstanty $C_1<C_2$ takové, že pro $n\in M$ máme $C_1n^{\log_b(a)}\leq f(n)\leq C_2n^{\log_b(a)}$. Pro $n=b^k$ to znamená, že $C_1a^k\leq f(b^k)\leq C_2a^k$.

Nyní tento odhad rozšíříme i na čísla v mezerách množiny $M$. Připomeňme, že pro $b^k$ máme $a^k=(b^k)^{\log_b(a)}$.

Vezměme libovolné $n\in\mathbb{N}$, $n\geq b$. Pak existuje $k\in\mathbb{N}$ takové, že $b^k\leq n<b^{k+1}$, a můžeme odhadovat následovně:

$$f(n)\leq f(b^{k+1})\leq C_2a^{k+1}=aC_2a^k=aC_2(b^k)^{\log_b(a)}\leq aC_2n^{\log_b(a)},$$
$$f(n)\geq f(b^k)\geq C_1a^k=\frac{C_1}{a}a^{k+1}=\frac{C_1}{a}(b^{k+1})^{\log_b(a)}\geq\frac{C_1}{a}n^{\log_b(a)}.$$

Označme $D_1=\frac{C_1}{a}$ a $D_2=aC_2$. Právě jsme dokázali, že pro $n\in\mathbb{N}$, $n\geq b$ platí $D_1n^{\log_b(a)}\leq f(n)\leq D_2n^{\log_b(a)}$. Odtud už plyne, že podobný odhad (jen možná s jinými konstantami) platí i pro všechna $n\in\mathbb{N}$. Máme $f(n)=\Theta(n^{\log_b(a)})$.

(ii): Jestliže $a=b^d$, tak podle příkladu 10c.c máme pro $n\in M$ vzorec $f(n)=f(1)n^d+cn^d\log_b(n)$. Druhý člen pro velká $n$ převáží a koeficient $c$ je kladný, proto existují konstanty $C_1<C_2$ takové, že pro $n\in M$ máme $C_1n^d\log_b(n)\leq f(n)\leq C_2n^d\log_b(n)$. Pro $n=b^k$ to tedy znamená, že $C_1b^{dk}k\leq f(d^k)\leq C_2b^{dk}k$.

Tento odhad zase rozšíříme. Použijeme přitom s úspěchem ekvivalentních nerovností $k+1\leq 2k$ a $k\geq\frac{k+1}{2}$, které evidentně platí pro $k\geq 1$.

Vezměme tedy libovolné $n\in\mathbb{N}$, $n\geq b$. Pak existuje $k\in\mathbb{N}$ takové, že $b^k\leq n<b^{k+1}$, a můžeme odhadovat takto:

$$f(n)\leq f(b^{k+1})\leq C_2b^{d(k+1)}(k+1)\leq C_2b^{dk}b^d2k=2b^dC_2(b^k)^dk\leq 2b^dC_2n^d\log_b(n),$$
$$f(n)\geq f(b^k)\geq C_1b^{dk}k\geq\frac{C_1}{b^d}b^{d(k+1)}\frac{k+1}{2}=\frac{C_1}{2b^d}(b^{k+1})^d(k+1)\geq\frac{C_1}{2b^d}n^d\log_b(n).$$

Označme $D_1=\frac{C_1}{2b^d}$ a $D_2=2b^dC_2$. Dokázali jsme, že pro $n\in\mathbb{N}$, $n\geq b$ je $D_1n^d\log_b(n)\leq f(n)\leq D_2n^d\log_b(n)$, tedy $f(n)=\Theta(n^d\log_b(n))$.

(iii): Jestliže $a<b^d$, tak podle příkladu 10c.c máme pro $n\in M$ vzorec $f(n)=c_1n^{\log_b(a)}+c_2n^d$, kde $c_1,c_2\in\mathbb{R}$ jsou konstanty nezávisející na $n$. Z $a<b^d$ máme $\log_b(a)<d$, proto druhý sčítanec převáží nad prvním, pokud tedy není násoben nulou. Máme $c_2=\dfrac{cb^d}{b^d-a}$, z předpokladů $c>0$, $b>0$ a $a<b^d$ tedy dostáváme $c_2>0$ a funkce $f(n)$ opravdu roste jako $n^d$. Proto existují konstanty $C_1<C_2$ takové, že pro $n\in M$ máme $C_1n^d\leq f(n)\leq C_2n^d$. Pro $n=b^k$ to tedy znamená, že $C_1b^{kd}\leq f(b^k)\leq C_2b^{kd}$.

Zase toto rozšíříme. Vezměme libovolné $n\in\mathbb{N}$, $n\geq b$. Pak existuje $k\in\mathbb{N}$ takové, že $b^k\leq n<b^{k+1}$, a můžeme odhadovat následovně:

$$f(n)\leq f(b^{k+1})\leq C_2b^{(k+1)d}=b^dC_2b^{kd}=b^dC_2(b^k)^d\leq b^dC_2n^d,$$
$$f(n)\geq f(b^k)\geq C_1b^{kd}=\frac{C_1}{b^d}b^{kd+d}=\frac{C_1}{b^d}(b^{k+1})^d\geq\frac{C_1}{b^d}n^d.$$

Označíme-li $D_1=\frac{C_1}{b^d}$ a $D_2=b^dC_2$, tak jsme právě dokázali, že pro $n\in\mathbb{N}$, $n\geq b$ platí $D_1n^d\leq f(n)\leq D_2n^d$ a tedy $f(n)=\Theta(n^d)$.

□

Interpretace: Začneme tím, že je-li funkce $f$ dána nejjednodušším homogenním vztahem $f(n)=a\cdot f\bigl(\frac{n}{b}\bigr)$, pak má růst $\Theta(n^{\log_b(a)})$, viz příklad 10c.b. Pokud něco na pravou stranu přidáme, pak nám důsledek tohoto kroku odhaluje právě dokázaná věta. Porovnává přitom $a$ s $b^d$, což je ekvivalentní s porovnáním $\log_b(a)$ a $d$. Věta tedy porovnává „přirozený růst“ s tím, co jsme přidali. To je velmi praktický pohled na věc, proto si tak Větu přepíšeme a zároveň do závěru zahrneme i onen homogenní případ, který jsme ve větě předpokladem $c > 0$ vyloučili (viz poznámka na konci příkladu 10c.c).

!

**Důsledek 10c.4.**
Předpokládejme, že nezáporná neklesající funkce $f$ na $\mathbb{N}$ splňuje rovnici $f(n) = a \cdot f\left(\frac{n}{b}\right) + cn^d$ na množině $M = \{b^k;\ k \in \mathbb{N}\}$, kde $b \in \mathbb{N}$ splňuje $b \geq 2$ a $a, c \in \mathbb{R}$, $d \in \mathbb{N}_0$ jsou konstanty splňující $a \geq 1$ a $c \geq 0$. Pak platí následující:
(i) Jestliže $d < \log_b(a)$ nebo $c = 0$, tak $f(n)$ je $\Theta(n^{\log_b(a)})$.
(ii) Jestliže $d = \log_b(a)$, tak $f(n)$ je $\Theta(n^{\log_b(a)} \log_2(n)) = \Theta(n^d \log_2(n))$.
(iii) Jestliže $d > \log_b(a)$, tak $f(n)$ je $\Theta(n^d)$.

Vidíme tedy, že pokud na pravou stranu nepřidáme moc (malá mocnina), tak se nic nestane, funkce roste stejně rychle, jako kdybychom na pravou stranu nepřidali nic. Převažuje tedy příspěvek od homogenní rovnice. Jakmile ale přesáhneme určitou mez, tak zcela převáží to, co jsme na pravou stranu přidali. Zajímavý je onen okamžik přechodu, výsledný vzorec je rychlejší než jak příspěvek od homogenní rovnice, tak příspěvek od pravé strany (jsou stejné), jako by se navzájem posilovali.

Při zkoumání algoritmů nám tedy po nalezení vhodného rekurentního vztahu věta 10c.3 (či její důsledek) okamžitě a bez jakékoliv další práce dává přesně to, co nás zajímá. Její název je tedy případný.

**Příklad 10c.d** (pokračování ): Vrátíme se k binárnímu vyhledávání. Odvodili jsme rovnici $f(n) = f\left(\frac{n}{2}\right) + 2$. Parametry jsou $a = 1$, $b = 2$ a $d = 0$, neboli $a = 1 = b^d$, máme také $\log_b(a) = \log_2(1) = 0$ takže nám Věta 10c.3 dává, že náročnost tohoto algoritmu je $f(n) = \Theta(n^0 \log_2(n)) = \Theta(\log_2(n))$. Odpovídá to tomu, co jsme si sami předtím intuitivně odvodili. Důsledek 10c.4 nám to samozřejmě dá také, tam bychom použili test $d = 0 = \log_2(1)$.
△

**Příklad 10c.e:** Zde si představíme rychlé násobení: Mějme dvě čísla, $a$ a $b$ o $n$ cifrách, řekněme v binárním tvaru. Standardní aloritmus pro násobení vyžaduje více než $n^2$ operací: násobení každého s každým pro jednotlivé bity obou čísel plus věci jako sčítání mezivýsledků, které náročnost dále zhorší, ale ne natolik, aby to zvýšilo rychlost růstu $n^2$. Mimochodem, toto klasické násobení je zase dosti úsporné na paměť, vyžaduje navíc jen cca $\log(n)$ registrů.

Existuje zajímavá finta: Rozdělíme obě čísla na poloviny (ve smyslu řetězců číslic) o délce $m = \frac{n}{2}$ míst, takže $a = A_1 \cdot 2^m + A_2$, $b = B_1 \cdot 2^m + B_2$. Pak

$$ab = (A_1 2^m + A_2)(B_1 2^m + B_2) = (2^{2m} + 2^m)A_1 B_1 + 2^m(A_1 - A_2)(B_2 - B_1) + (2^m + 1)A_2 B_2$$

(ověřte). Všimněte si, že se tam násobí jen čísla velikosti $m$, a to třikrát, čili náročnost je $3m^2 = \frac{3}{4}n^2$. Jsou tam i další operace, ale ty jsou ve srovnání s $\frac{3}{4}n^2$ nenáročné. Vypadají sice na první pohled jako násobení, jenže pokud jsou všechna uvažovaná čísla v binárním tvaru, pak jde vlastně jen o posuny doleva, což jsou velice rychlé operace ve srovnání s násobením.

Vidíme, že tímto půlícím trikem jsme se dostali z $n^2$ na $\frac{3}{4}n^2$, což je pokrok, ovšem nic nám nebrání podobnou fintu rekurzivně aplikovat i na ta tři násobení, dostaneme tak algoritmus pro chytré násobení čísel délky $n = 2^k$. Kolik zabere operací?

Když započítáme doplňující faktory, které jsou všechny lineárně náročné, dostaneme rovnici $f(n) = 3f\left(\frac{n}{2}\right) + cn$. Máme konstanty $a = 3$, $b = 2$ a $d = 1$, zde je $d$ je menší než $\log_b(a) = \log_2(3)$ a podle Důsledku 10c.4 je tedy náročnost algoritmu řádu $\Theta(n^{\log_2(3)})$. Protože $\log_2(3) \sim 1.585...$, je to lepší než klasické násobení s jeho $n^2$.

Poznámka: Podobná finta existuje pro násobení matic, které standardně „stojí“ $n^3$ násobení a $n^2(n-1)$ sčítání, čili v zásadě je to algoritmus s náročností $n^3$. Chytré násobení pro matice je založeno na rovnosti, pomocí které se místo jednoho násobení dvou $n \times n$ matic sedmkrát násobí dvě matice o rozměru $\frac{n}{2}$ a pak se ještě použije 15 sčítání matic téže velikosti, zase se to dá zrekurzivnit a dostaneme $f(n) = 7f\left(\frac{n}{2}\right) + 15\left(\frac{n}{2}\right)^2$. Máme tedy $a = 7$, $b = 2$, $d = 2$, opět je $d$ menší než $\log_b(a) = \log_2(7) \sim 2.8$, proto má toto maticové násobení náročnost $\Theta(n^{\log_2(7)})$, což je poněkud lepší než obvyklých $n^3$.
△

Podobně jako u lineárních rekurentních rovnic je také možné pravé strany kombinovat.

! **Věta 10c.5.** (o superpozici)
Nechť $a \in \mathbb{R}$, $b \in \mathbb{N}$ splňují $a \geq 1$ a $b \geq 2$, označme $M = \{b^k;\ k \in \mathbb{N}\}$.
Jestliže funkce $f_1$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_1(n)$ pro všechna $n \in M$
a funkce $f_2$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_2(n)$ pro všechna $n \in M$,
pak funkce $f_1 + f_2$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_1(n) + g_2(n)$ pro všechna $n \in M$.

Indukcí se to samozřejmě dá snadno rozšířit na libovolný (konečný) počet sčítaných funkcí $g$ na pravé straně. Co z toho pro nás plyne prakticky? Jestliže zkoumáme funkci danou vztahem $f(n) = af\big(\frac{n}{b}\big) + p(n)$, kde $p$ je polynom, tak pro každou mocninu tohoto polynomu dostaneme funkci známého růstu. Když je sečteme, dostaneme řešení celé dané rovnice, a to roste tak rychle, jak rychle roste nejrychlejší ze sčítanců. Podíváme-li se na jednotlivé možnosti u Důsledku 10c.4, tak hned vidíme, že když dohromady zamícháme řešení odpovídající několika hodnotám $d$, tak nejrychleji z nich poroste právě to, které odpovídá největšímu $d$. Z toho vyplývá následující závěr.

**Praktické pravidlo:** Jestliže je funkce $f$ určena rovnicí $f(n) = af\big(\frac{n}{b}\big) + p(n)$, kde $p$ je polynom, pak je rychlost růstu $f$ určena podle Důsledku 10c.4, kde za $d$ vezmeme stupeň polynomu $p$.

Jinými slovy, je-li na pravé straně polynom, tak záleží jen na jeho největší mocnině (což je vlastně při úvahách s rychlostí staré dobré pravidlo, viz kapitola 9b).

Pozorný čtenář ovšem může namítnout, že jsme se počínaje Větou 10c.3 omezili na $c \geq 0$. Co se stane, když budou některé (všechny) koeficienty v polynomu záporné? Z praktického pohledu to není problém. Buď je příslušná mocnina tak malá, aby výsledek neovlivnila ($d < \log_b(a)$), pak záporný koeficient nevadí, protože skutečný růst funkce udává jiná část vztahu. Nebo je záporný koeficient u mocniny, která je dost velká na to, aby určila růst $f$, pak se dozvíme třeba to, že $f$ je jako $-n^2$, což je ale u praktických příkladů nemožné, neboť náročnost algoritmů nemůže být záporná. Čili tam, kde by záporné koeficienty mohly vadit, se zase u prakticky motivovaných úloh nemohou objevit.

Na závěr jednu kuriozitu. Rovnice typu $T(n) = aT^2(n/b)$ díky té druhé mocnině neumíme zkoumat pomocí výsledků této kapitoly. Nabízí se trik. Nejprve použijeme substituci $n = b^k$, dostáváme $T(b^k) = aT^2(b^{k-1})$. Teď rovnici zlogaritmujeme: $\ln(T(b^k)) = \ln(a) + 2\ln(T(b^{k-1}))$. Když označíme $a_k = \ln(T(b^k))$, dostaneme rovnici $a_k = 2a_{k-1} + \ln(a)$, což je lineární rekurentní rovnice a snadno ji vyřešíme pomocí algoritmu 10b.8.

Co to ukazuje? Učebnice se většinou zabývají výkladem metod pro řešení určitých typů rovnic, takže by to mohlo vzbudit dojem, že rovnice umíme nějak plánovitě řešit. Ve skutečnosti umíme takto řešit jen velice malou skupinku rovnic, těch nejpěknějších. Stačí malá modifikace a objeví se rovnice, na kterou obvyklé metody nelze aplikovat. Pak to začne být zajímavé, znalost metod je jen nezbytný základ, ale pak přichází hledání fint a triků, jak si poradit s něčím, co do našich škatulek nezapadá.

## Cvičení

**Cvičení 10c.1** (rutinní)**:** (i) Uvažujte funkci danou $f(n) = f\big(\frac{n}{4}\big) + 3$ a $f(1) = 1$. Spočítejte $f(4)$, $f(16)$, $f(256)$.
(ii) Uvažujte funkci danou $f(n) = f\big(\frac{n}{2}\big) + n^2$ a $f(1) = 0$. Spočítejte $f(2)$, $f(4)$ a $f(32)$.

**Cvičení 10c.2** (rutinní, poučné, *zkouškové)**:** Pro následující funkce nejprve odhadněte přesný vzorec na množině $M = \{b^k\}$ iterací definičního vztahu a dokažte indukcí jeho správnost (viz příklad 10c.b, popř. důkaz Lemma 10c.2), poté aplikujte Master theorem (či jeho důsledek) k ověření asymptotické rychlosti růstu funkce.
(i)* $f(n) = 2f\big(\frac{n}{3}\big)$, $f(1) = 13$;
(ii)* $f(n) = f\big(\frac{n}{3}\big) + 1$, $f(1) = 13$;
(iii)* $f(n) = 2f\big(\frac{n}{2}\big) + 1$, $f(1) = 13$;
(iv)* $f(n) = 3f\big(\frac{n}{2}\big) + 1$, $f(1) = 13$;
(v) $f(n) = f\big(\frac{n}{2}\big) + \frac{n}{2}$, $f(1) = 13$;
(vi) $f(n) = f\big(\frac{n}{2}\big) + 3n^2$, $f(1) = 13$.

**Cvičení 10c.3** (rutinní)**:** Pro následující funkce určete rychlost jejich asymptotickou růstu pomocí Master theorem, popřípadě Důsledku 10c.4:
(i) $f(n) = 2f\big(\frac{n}{3}\big)$;
(ii) $f(n) = f\big(\frac{n}{3}\big)$;
(iii) $f(n) = 4f\big(\frac{n}{2}\big)$;
(iv) $f(n) = f\big(\frac{n}{3}\big) + 2n$;
(v) $f(n) = 3f\big(\frac{n}{3}\big) + 2n$;
(vi) $f(n) = 2f\big(\frac{n}{4}\big) + 27$;
(vii) $f(n) = 8f\big(\frac{n}{2}\big) + 13n$;
(viii) $f(n) = 4f\big(\frac{n}{2}\big) + n^2$;
(ix) $f(n) = 4f\big(\frac{n}{2}\big) + n^3$.

**Cvičení 10c.4** (poučné)**:** V tomto cvičení bude stručně popsáno několik užitečných algoritmů. Pro každý z nich sestavte rekurentní rovnici popisující náročnost algoritmu, poté odhadněte asymptotickou rychlost růstu dotyčné funkce pomocí Věty 10c.3 či Důsledku 10c.4.
(i) Chceme-li počítat mocninu $x^n$ přímo, vyžaduje $n - 1$ násobení.
Rychlé mocnění: Jestliže máme umocňovat na sudou mocninu, můžeme použít $x^{2m} = (x^m)^2$. K výpočtu je tedy potřeba 1) spočítat $x^m$ 2) vynásobit $x^m \cdot x^m$.

Pokud je i $m$ sudé, můžeme v rozkladu rekurzivně pokračovat, ideální je používat tento algoritmus na mocniny typu $x^{2^k}$. Určete, kolik je pak třeba násobení.
Například pro výpočet $x^8$ stačí násobit $x^2 = x \cdot x$, $x^4 = x^2 \cdot x^2$ a $x^8 = x^4 \cdot x^4$, tedy celkem třikrát.
Bonus: Takto lze zjednodušit výpočet libovolné mocniny. Například $x^{13}$ si napíšeme jako $x^{1+4+8} = x \cdot x^4 \cdot x^8$. Tři násobení nám dala všechny nutné mocniny typu $x^{2^k}$, další dvě násobení nám dají $x^{13}$. Stačí tedy pět násobení namísto dvanácti. Odhadněte počet násobení nutný k výpočtu obecného $x^m$.
(ii): Máme seznam o $n$ položkách a chceme jej seřadit podle velikosti/abecedy. Pokud bychom použili metodu „najdi největší, pak najdi největší ze zbytku, pak najdi největší ze zbytku…“, bylo by třeba řádově $n^2$ porovnání.
Merge sort: Nechť má seznam sudý počet položek $n = 2m$. Rozdělíme jej na poloviny, uspořádáme každou z nich a dva uspořádané seznamy o délce $m$ spojíme do jednoho uspořádaného seznamu, což lze udělat s pouhými $2m = n$ srovnáními. Pro $n = 2^k$ lze postup snadno iterovat. Odhadněte počet srovnání potřebný k urovnání celého seznamu.
(iii) Máme seznam o $n$ položkách a chceme najít největší a nejmenší položku v seznamu. To stojí přinejhorším $2n$ srovnání, pokud použijeme přímočarý útok: První číslo si schováme jako min a max, každé další číslo porovnáme s dočasným minimem a maximem a pokud je extrémnější, nahradíme jím příslušnou hodnotu.
Pokus o nápodobu merge sortu: Rozdělíme seznam na polovinu, najdeme u každé poloviny její maximum a minimum, pak stačí porovnat obě maxima a obě minima a máme globální extrémy. Kolik porovnání vyžaduje algoritmus, když toto půlení iterujeme?

**Cvičení 10c.5** (dobré, poučné)**:** Předpokládejme, že funkce $f$ splňuje rekurentní vztah $f(n) = 3f\big(\sqrt{n}\big) + 13$ pro $n$ ve tvaru $k^2$, $k \in \mathbb{N}$, $k \geq 2$ a platí $f(2) = 1$.
(i) Najděte $f(16)$
(ii) Uvažujte funkci $F(m) = f(2^m)$. Pomocí původní rekurentní rovnice najděte novou rekurentní rovnici pro $F$, určete rychlost růsu $F$ a pak i $f$.

**Cvičení 10c.6** (dobré, poučné)**:** Nechť $a \in \mathbb{R}$, $b \in \mathbb{N}$ splňují $a \geq 1$ a $b \geq 2$, označme $M = \{b^k;\ k \in \mathbb{N}\}$. Dokažte následující:
Jestliže funkce $f_1$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_1(n)$ pro všechna $n \in M$
a funkce $f_2$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_2(n)$ pro všechna $n \in M$,
pak funkce $f_1 + f_2$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g_1(n) + g_2(n)$ pro všechna $n \in M$.

**Cvičení 10c.7** (dobré, poučné)**:** Nechť $a \in \mathbb{R}$, $b \in \mathbb{N}$ splňují $a \geq 1$ a $b \geq 2$, označme $M = \{b^k;\ k \in \mathbb{N}\}$. Nechť funkce $f_p$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g(n)$ pro všechna $n \in M$.
Dokažte, že funkce $f$ splňuje rovnici $f(n) = af\big(\frac{n}{b}\big) + g(n)$ pro všechna $n \in M$ právě tehdy, když $f = f_p + f_h$, kde $f_h$ je nějaká funkce splňující rovnici $f(n) = af\big(\frac{n}{b}\big)$ pro všechna $n \in M$.

**Cvičení 10c.8** (dobré, poučné)**:** Nechť $a \in \mathbb{R}$, $b \in \mathbb{N}$ splňují $a \geq 1$ a $b \geq 2$, označme $M = \{b^k;\ k \in \mathbb{N}\}$. Nechť $N$ je množina všech funkcí na $M$ splňujících rovnici $f(n) = af\big(\frac{n}{b}\big)$ pro všechna $n \in M$.
Dokažte, že $N$ je jednodimenzionální vektorový prostor.

**Řešení:**
**10c.1:** (i): $f(4) = f(1) + 3 = 1 + 3 = 4$, $f(16) = f(4) + 3 = 4 + 3 = 7$, pracovní mezivýsledek $f(64) = f(16) + 3 = 7 + 3 = 10$ a tedy $f(256) = f(64) + 3 = 10 + 3 = 13$.
(ii): $f(2) = f(1) + 2^2 = 0 + 4 = 4$, $f(4) = f(2) + 4^2 = 4 + 16 = 20$, pracovní mezivýsledky $f(8) = f(4) + 8^2 = 20 + 64 = 84$, $f(16) = f(8) + 16^2 = 84 + 256 = 340$ a tedy $f(32) = f(16) + 32^2 = 340 + 1024 = 1364$.
**10c.2:** (i): Přepis: $f(3n) = 2f(n)$; $f(3) = 2f(1) = 2 \cdot 13$, $f(3^2) = 2f(3) = 2 \cdot (2 \cdot 13) = 2^2 \cdot 13$,
$f(3^3) = 2f(3^2) = 2 \cdot (2^2 \cdot 13) = 2^3 \cdot 13$, odhad $f(3^k) = 13 \cdot 2^k$.
(0) $k = 0$: $f(3^0) = 13 \cdot 2^0 = 13 = f(1)$.
(1) $f(3^k) = 13 \cdot 2^k \implies f(3^{k+1}) = 2f(3^k) = 13 \cdot 2^{k+1}$ souhlasí.
Přepis: $f(n) = 13 \cdot 2^{\log_3(n)} = 13 \cdot n^{\log_3(2)}$ na $M$.
Důsledek 10c.4: $a = 2$, $b = 3$, $c = 0$, proto $f(n) = \Theta(n^{\log_3(2)})$.
(ii): Přepis: $f(3n) = f(n) + 1$; $f(3) = f(1) + 1 = 13 + 1$, $f(3^2) = f(3) + 1 = (13 + 1) + 1 = 13 + 2$,
$f(3^3) = f(3^2) + 1 = (13 + 2) + 1 = 13 + 3$, odhad $f(3^k) = 13 + k$.
(0) $k = 0$: $f(3^0) = 13 + 0 = 13 = f(1)$.
(1) $f(3^k) = 13 + k \implies f(3^{k+1}) = f(3^k) + 1 = (13 + k) + 1 = 13 + (k + 1)$ souhlasí.
Přepis: $f(n) = 13 + \log_3(n) = 13 + \log_3(2)\log_2(n)$ na $M$.
Důsledek 10c.4: $a = 1$, $b = 3$, $d = 0 = \log_3(1)$, proto $f(n) = \Theta(n^0 \log_2(n)) = \Theta(\log_2(n))$.
(iii): Přepis: $f(2n) = 2f(n) + 1$; $f(2) = 2f(1) + 1 = 2 \cdot 13 + 1$, $f(2^2) = 2f(2) + 1 = 2 \cdot (2 \cdot 13 + 1) + 1 = 2^2 \cdot 13 + 1 + 2$,
$f(2^3) = 2f(2^2) + 1 = 2 \cdot (2^2 \cdot 13 + 1 + 2) + 1 = 2^3 \cdot 13 + 1 + 2 + 4$,

$f(2^4)=2f(2^3)+1=2\cdot(2^3\cdot 13+1+2+4)+1=2^4\cdot 13+1+2+4+8$,
odhad $f(2^k)=13\cdot 2^k+1+2+4+\dots+2^{k-1}=13\cdot 2^k+\frac{1-2^k}{1-2}=13\cdot 2^k+2^k-1=14\cdot 2^k-1$.
(0) $k=0$: $f(2^0)=14\cdot 2^0-1=13=f(1)$.
(1) $f(2^k)=14\cdot 2^k-1\implies f(2^{k+1})=2f(2^k)+1=14\cdot 2^{k+1}-2+1=14\cdot 2^{k+1}-1$ souhlasí.
Přepis: $f(n)=14n-1=$ na $M$.
Důsledek 10c.4: $a=2$, $b=2$, $d=0<\log_2(2)=1$, proto $f(n)=\Theta(n^{\log_2(2)})=\Theta(n)$.
(iv): Přepis: $f(2n)=3f(n)+1$; $f(2)=3f(1)+1=3\cdot 13+1$, $f(2^2)=3f(2)+1=3\cdot(3\cdot 13+1)+1=3^2\cdot 13+1+3$,
$f(2^3)=3f(2^2)+1=3\cdot(3^2\cdot 13+1+3)+1=3^3\cdot 13+1+3+9$,
$f(2^4)=3f(2^3)+1=3\cdot(3^3\cdot 13+1+3+9)+1=3^4\cdot 13+1+3+9+27$,
odhad $f(2^k)=13\cdot 3^k+1+3+9+\dots+3^{k-1}=13\cdot 3^k+\frac{1-3^k}{1-3}=13\cdot 3^k+\frac{1}{2}\cdot 3^k-\frac{1}{2}=\big(13+\frac{1}{2}\big)\cdot 3^k-\frac{1}{2}$.
(0) $k=0$: $f(2^0)=\big(13+\frac{1}{2}\big)\cdot 2^0-\frac{1}{2}=13=f(1)$.
(1) $f(2^k)=\big(13+\frac{1}{2}\big)\cdot 3^k-\frac{1}{2}\implies f(2^{k+1})=3f(2^k)+1=\big(13+\frac{1}{2}\big)\cdot 3^{k+1}-\frac{3}{2}+1=\big(13+\frac{1}{2}\big)\cdot 3^{k+1}-\frac{1}{2}$ souhlasí.
Přepis: $f(n)=\big(13+\frac{1}{2}\big)\cdot 3^{\log_2(n)}-\frac{1}{2}=\big(13+\frac{1}{2}\big)n^{\log_2(3)}-\frac{1}{2}$ na $M$.
Důsledek 10c.4: $a=3$, $b=2$, $d=0<\log_2(3)$, proto $f(n)=\Theta(n^{\log_2(3)})$.
(v): Přepis: $f(2n)=f(n)+n$; $f(2)=f(1)+1=13+1$, $f(2^2)=f(2)+2=13+1+2$,
$f(2^3)=f(2^2)+2^2=13+1+2+2^2$, $f(2^4)=f(2^3)+2^3=13+1+2+2^2+2^3$,
odhad $f(2^k)=13+1+2+4+\dots+2^{k-1}=13+\frac{1-2^k}{1-2}=13+2^k-1=2^k+12$.
(0) $k=0$: $f(2^0)=2^0+12=13=f(1)$.
(1) $f(2^k)=2^k+12\implies f(2^{k+1})=f(2^k)+2^k=2^k+12+2^k=2\cdot 2^k+12=2^{k+1}+12$ souhlasí.
Přepis: $f(n)=n+12$ na $M$.
Důsledek 10c.4: $a=1$, $b=2$, $d=1>\log_2(1)=0$, proto $f(n)=\Theta(n)$.
(vi): Přepis: $f(2n)=f(n)+3(2n)^2=f(n)+12n^2$; $f(2)=f(1)+12\cdot 1^2=13+12\cdot 1$,
$f(2^2)=f(2)+12\cdot 2^2=13+12\cdot 1+12\cdot 2^2$, $f(2^3)=f(2^2)+12\cdot(2^2)^2=13+12\cdot 1+12\cdot 2^2+12\cdot(2^2)^2$,
$f(2^4)=f(2^3)+12\cdot(2^3)^2=13+12\cdot 1+12\cdot 2^2+12\cdot(2^2)^2+12\cdot(2^2)^3$,
odhad $f(2^k)=13+12(1+4+4^2+\dots+4^{k-1})=13+12\frac{1-4^k}{1-4}=13+4\cdot 4^k-4=4^{k+1}+9$.
(0) $k=0$: $f(2^0)=4^1+9=13=f(1)$.
(1) $f(2^k)=4^{k+1}+9\implies f(2^{k+1})=f(2^k)+12(2^k)^2=4^{k+1}+9+12(2^2)^k=4^{k+1}+3\cdot 4^{k+1}+9=4\cdot 4^{k+1}+9=4^{k+2}+9$ souhlasí.
Přepis: $f(n)=4^{\log_2(n)+2}+9=(2^2)^{\log_2(n)}\cdot 4^2+9=(2^{\log_2(n)})^2\cdot 16+9=16n^2+9$ na $M$.
Důsledek 10c.4: $a=1$, $b=2$, $d=2>\log_2(1)=0$, proto $f(n)=\Theta(n^2)$.
**10c.3:** (i): $a=2$, $b=3$, $c=0$, tedy $f(n)=\Theta(n^{\log_3(2)})$;
(ii): $a=1$, $b=3$, $c=0$, tedy $f(n)=\Theta(n^{\log_3(1)})=\Theta(1)$;
(iii): $a=4$, $b=2$, $c=0$, tedy $f(n)=\Theta(n^{\log_2(4)})=\Theta(n^2)$;
(iv): $a=1$, $b=3$, $d=1>\log_3(1)=0$, tedy $f(n)=\Theta(n)$;
(v): $a=3$, $b=3$, $d=1=\log_3(3)$, tedy $f(n)=\Theta(n\log_2(n))$;
(vi): $a=2$, $b=4$, $d=0<\log_4(2)=\frac{1}{2}$, tedy $f(n)=\Theta(n^{\log_4(2)})=\Theta\big(\sqrt{n}\,\big)$;
(vii): $a=8$, $b=2$, $d=1<\log_2(8)=3$, tedy $f(n)=\Theta(n^{\log_2(8)})=\Theta(n^3)$;
(viii): $a=4$, $b=2$, $d=2=\log_2(4)$, tedy $f(n)=\Theta(n^{\log_2(4)}\log_2(n))=\Theta(n^2\log_(N))$;
(ix): $a=4$, $b=2$, $d=3>\log_2(4)$, tedy $f(n)=\Theta(n^3)$.
**10c.4:** (i): $f(n)=f\big(\frac{n}{2}\big)+1$, $a=1$, $b=2$, $d=0=\log_2(1)$, proto $f(n)=\Theta(n^0\log_2(n))=\Theta(\log_2(n))$.
Poznámka: Je to rozhodně lepší než $n$ při umocňování podle definice.
Bonus: Každé $m\in\mathbb{N}$ lze napsat jako součet mocnin typu $2^i$ (neboli zapsat ve dvojkové soustavě), nejvyšší použitá mocina $2^k$ je dána jako $k=\lfloor\log_2(m)\rfloor$. Mocninu $x^m$ pak získáme vynásobením těch z mocnin $x^1$, $x^2$, $x^4$, …, $x^{2^k}$, které se objeví v rozkladu. Výpočet nejvyšší mocniny nás dle předchozího výpočtu stojí $\log_2(2^k)=k$ násobení, těch mocnin je celkem $k+1$ a v nejhorším násobíme všechny, což je dalších $k$ násobení. Nejhorší scénář pro výpočet $x^m$ tímto způsobem tedy dává $2k=2\lfloor\log_2(m)\rfloor$ násobení, tedy náročnost $\Theta(\log_2(m))$ násobení. To je velmi pěkné.
(ii): $f(n)=2f\big(\frac{n}{2}\big)+n$, $a=2$, $b=2$, $d=1=\log_2(2)$, proto $f(n)=\Theta(n^1\log_2(n))=\Theta(n\log_2(n))$.
Poznámka: Je to dost lepší než oněch $n^2$.
(iii): $f(n)=2f\big(\frac{n}{2}\big)+2$, $a=2$, $b=2$, $d=0<\log_2(2)=1$, proto $f(n)=\Theta(n^{\log_2(2)})=\Theta(n)$.
Poznámka: Je to tedy řádově stejně rychlé jako přímý útok, moc jsme si nepomohli.
**10c.5:** (i): $f(4)=3f(2)+13=3\cdot 1+13=16$, $f(16)=3f(4)+13=3\cdot 16+13=61$.
(ii): $F(m)=f(2^m)=3f(2^{m/2})+13=3F\big(\frac{m}{2}\big)+13$. Důsledek 10c.4: $a=3$, $b=2$, $d=0<\log_2(3)$, proto $F(m)=\Theta(m^{\log_2(3)})$, to dává $f(2^m)=\Theta(m^{\log_2(3)})$ a proto $f(n)=\Theta([\log_2(n)]^{\log_2(3)})$.
**10c.6:** Dosazení: $(f_1+f_2)(n)=f_1(n)+f_2(n)=af_1\big(\frac{n}{b}\big)+g_1(n)+af_2\big(\frac{n}{b}\big)+g_2(n)=a(f_1+f_2)\big(\frac{n}{b}\big)+(g_1(n)+g_2(n))$.
**10c.7:** 1) Nechť $f$ je řešení. Definujte $f_h=f-f_p$, dosadit:

$f_h(n) = f(n) - f_p(n) = af\big(\frac{n}{b}\big) + g(n) - af_p\big(\frac{n}{b}\big) - g(n) = a(f - f_p)\big(\frac{n}{b}\big) = af_h)\big(\frac{n}{b}\big)$.
2) Nechť $f_h$ je homogenní řešení. Pak podle věty o superpozici $f_p + f_h$ řeší rovnici $f(n) = af\big(\frac{n}{b}\big) + g(n) + 0$.
**10c.8:** Podle příkladu 10c.b je $N = \{f(b^k) = a^k f(1)\} = \{f(b^k) = c \cdot a^k;\ c \in \mathbb{R}\}$, jde tedy o jednorozměrný vektorový prostor s bází danou funkcí $f(b^k) = a^k$.

## 10d. Bonus: Generující funkce

Zde si představíme zajímavou alternativní metodu řešení rekurentních rovnic. Bude založena na materiálu vyloženém v kapitole 9d, kde jsme se seznámili s řadami a mocninnými řadami. Začneme tím, že se na ty věci podíváme trochu z jiné strany.

Je jasné, že existuje vzájemně jednoznačně vztah mezi posloupnostmi a mocninnými řadami. Každá posloupnost $\{a_k\}_{k=0}^{\infty}$ dává vzniknout odpovídající mocninné řadě $\sum\limits_{k=0}^{\infty} a_k x^k$ a naopak, pokud si z mocninné řady vytáhneme její koeficienty $a_k$, vytvoří nám posloupnost. Nás budou následně zajímat jen řady, které se chovají rozumně vzhledem ke konvergenci. Uvažujme proto množinu $M$ všech posloupností $\{a_k\}_{k=0}^{\infty}$ takových, že pro ně odpovídající řady $\sum\limits_{k=0}^{\infty} a_k x^k$ konvergují na nějakém nedegenerovaném intervalu, jinými slovy, mají poloměr konvergence $\varrho > 0$.

Pokud má nějaká řada netriviální obor konvergence, tak na tomto intervalu definuje jistou funkci. To znamená, že vlastně ke každé posloupnosti z $M$ dostáváme určitou funkci $f$, která je definovaná na nějakém intervalu $(-\varrho, \varrho)$ pro $\varrho > 0$. Vzniká nám tím přiřazení neboli zobrazení. Označme jako $N$ množinu všech funkcí, které dostaneme pomocí posloupností z $M$, nechť $T$ je příslušné zobrazení $M \mapsto N$, které ke každé posloupnosti $\{a_k\}_{k=0}^{\infty}$ přiřadí funkci danou jako součet mocninné řady $\sum\limits_{k=0}^{\infty} a_k x^k$.

**Příklad 10d.a:** Víme, že $\sum\limits_{k=0}^{\infty} x^k = \frac{1}{1-x}$ na $(-1, 1)$. Tato řada se dá napsat jako $\sum\limits_{k=0}^{\infty} 1 \cdot x^k$ neboli odpovídá posloupnosti $\{1\}_{k=0}^{\infty} = (1, 1, 1, 1, \dots)$.

Můžeme tedy tvrdit, že $\{1\}_{k=0}^{\infty} \in M$ a $T(\{1\}) = \frac{1}{1-x}$.

Další známý rozvoj je pro exponenciálu, $e^x = \sum\limits_{k=0}^{\infty} \frac{x^k}{k!}$. Vidíme tedy, že $\big\{\frac{1}{k!}\big\} \in M$ a $T\big(\big\{\frac{1}{k!}\big\}\big) = e^x$.

Naopak se pomocí analytických metod snadno ukáže, že řada $\sum\limits_{k=0}^{\infty} k!x^k$ konverguje jen pro $x = 0$, tedy má poloměr konvergence $\varrho = 0$. Proto posloupnost $\{k!\}_{k=0}^{\infty}$ neleží v $M$ a v této teorii na ni tedy nedosáhneme.
△

Jaké vlastnosti můžeme od našich nových pojmů čekat? Hodně příjemné a občas také zajímavé.

**Věta 10d.1.**
Množina $M$ je lineární prostor.
Zobrazení $T$ je lineární.

**Důkaz** (poučný)**:** Vezměme dvě posloupnosti $\{a_k\}_{k=0}^{\infty}, \{b_k\}_{k=0}^{\infty} \in M$. Odpovídající řady pak konvergují na netriviálním intervalu, označme funkce odpovídající příslušným řadám jako $f(x) = \sum\limits_{k=0}^{\infty} a_k x^k$, $g(x) = \sum\limits_{k=0}^{\infty} b_k x^k$. V našem novém značení to znamená, že $f(x) = T(\{a_k\})$ a $g(x) = T(\{b_k\})$.

Součet posloupností je definován jako $\{a_k\}_{k=0}^{\infty} + \{b_k\}_{k=0}^{\infty} = \{a_k + b_k\}_{k=0}^{\infty}$ a této posloupnosti odpovídá řada $\sum\limits_{k=0}^{\infty} (a_k + b_k)x^k$. Věta 9d.4 říká, že tato řada konverguje a její součet je roven $\sum\limits_{k=0}^{\infty} a_k x^k + \sum\limits_{k=0}^{\infty} b_k x^k = f(x) + g(x)$. Tím se dozvídáme dvě věci. Za prvé, součet $\{a_k\}_{k=0}^{\infty} + \{b_k\}_{k=0}^{\infty}$ leží v $M$, tedy tato množina je uzavřená na sčítání. Za druhé, tomuto součtu odpovídá funkce $f(x) + g(x)$, což se dá zapsat jako $T(\{a_k\} + \{b_k\}) = T(\{a_k\}) + T(\{b_k\})$ a jedna podmínka linearity pro $T$ je splněna.

Nechť $c \in \mathbb{R}$. Posloupnosti $c\{a_k\}_{k=0}^{\infty} = \{ca_k\}_{k=0}^{\infty}$ odpovídá řada $\sum\limits_{k=0}^{\infty} ca_k x^k$. Podle Věty 9d.4 tato řada konverguje a její součet je $\sum\limits_{k=0}^{\infty} ca_k x^k = cf(x)$, proto $c\{a_k\}_{k=0}^{\infty} \in M$ a $T(\{ca_k\}) = cf(x) = cT(\{a_k\})$, čímž se potvrdila druhá podmínka linearity $T$.

Dokázali jsme, že množina $M$ je coby podmnožina lineárního prostoru všech posloupností uzavřená na sčítání a násobek konstantou, tudíž je $M$ také lineární prostor.

□

Linearita bude velice užitečná, díky ní už například hravě odvodíme, kam se posílají konstantní posloupnosti $\{a\}_{k=0}^{\infty} = (a, a, a, \dots)$:

$$T(\{a\}_{k=0}^{\infty}) = aT(\{1\}_{k=0}^{\infty}) = \frac{a}{1-x}.$$

Vztah mezi posloupností a funkcí je v některých aplikacích tak užitečný, že si zaslouží své jméno.

**Definice**
Řekneme, že funkce $f$ je **generující funkcí** (**generating function**) pro posloupnost $\{a_k\}_{k=0}^{\infty}$, jestliže na nějakém intervalu $(-\varrho, \varrho)$ pro $\varrho > 0$ platí $f(x) = \sum\limits_{k=0}^{\infty} a_k x^k$.

Někdy se tomu také říká **vytvořující funkce**.

My teď potřebujeme udělat dvě věci. Jednak si vytvořit nějaký slovníček výrazů, které umíme transformovat, a za druhé rozšířit vlastnosti, protože linearita nám nebude stačit, budeme chtít dělat i jiné triky.

Abychom si udělali rozumný slovníček, zamyslíme se nejprve nad vlastnostmi $T$ z trochu jiné strany. Již z definice je jasné, že $T$ je na. Otázka, zda je prosté, je ovšem značně těžká a vyžaduje to tvrdou analytickou práci. Nakonec se ale ukáže, že když máme řady s rozdílnými koeficienty, tak už nutně musí dávat různé funkce. To znamená, že $T$ je bijekce a tudíž můžeme pracovat i s její inverzí $T^{-1}$. Víme už například, že $T^{-1}\bigl(\frac{1}{1-x}\bigr) = \{1\}_{k=0}^{\infty}$. Tento oboustranný vztah budeme značit $\{1\} \leftrightarrow \frac{1}{1-x}$.

V kapitole 9d jsme ukázali ještě jeden součet řady, ze kterého dostaneme další obousměrný vztah.

**Fakt 10d.2.** (slovník)
$T(\{1\}_{k=0}^{\infty}) = \dfrac{1}{1-x}$ neboli $(1, 1, 1, 1, \dots) \leftrightarrow \dfrac{1}{1-x}$;
$T(\{k+1\}_{k=0}^{\infty}) = \dfrac{1}{(1-x)^2}$ neboli $(1, 2, 3, 4, \dots) \leftrightarrow \dfrac{1}{(1-x)^2}$.

Není to zrovna nejbohatší slovník, ale ve spojení s triky, které uvidíme vzápětí, nám to postačí. Dalším krokem jsou pravidla, která nám umožní pomocí slovníčku pracovat i s příbuznými výrazy. V zásadě jde jen o standardní manipulace s mocninnými řadami vyjádřené v našem novém jazyce. Nejprve si ukážeme jeden příklad, který snad ozřejmí, že opravdu nejde o nic jiného než dobrou práci s řadami, při které se často vyplatí si je napsat v dlouhém tvaru. To se nám bude hodit v důkazech, které přijdou.

**Příklad 10d.b:** Co víme o posloupnosti $(1, 0, 1, 0, 1, 0, 1, \dots)$? Odpovídá jí řada

$$x^0 + x^2 + x^4 + x^6 + \dots = \sum_{k=0}^{\infty} x^{2k} = \sum_{k=0}^{\infty} (x^2)^k = \frac{1}{1-x^2}.$$

Takže $(1, 0, 1, 0, 1, \dots) \in M$ a $T(1, 0, 1, 0, 1, \dots) = \frac{1}{1-x^2}$ neboli $(1, 0, 1, 0, 1, \dots) \leftrightarrow \frac{1}{1-x^2}$.

Dá se tato posloupnost zapsat nějak přesně? Nejjednodušší je specifikovat její členy coby $a_k = \begin{cases} 1, & k \text{ sudé}; \\ 0, & k \text{ liché}. \end{cases}$

Pokud chceme pěkný vzoreček, nabízí se trik, rozmyslete si, že to je vlastně posloupnost $\bigl\{\frac{1}{2}[1 + (-1)^k]\bigr\}_{k=0}^{\infty}$.
△

Teď už se podívejme na pravidla, připomeňme si, že už jsme dokázali linearitu. Co ještě můžeme chtít s posloupností udělat? Kromě násobení celé posloupnosti konstantou $\{ca_k\}$ je možné také tuto konstantu umocňovat, tedy z posloupnosti $\{a_k\}$ vyrobit $\{c^k a_k\}$. Občas se hodí umět přejít k posloupnosti $\{ka_k\}$. Poslední významnou skupinou operací je posun v posloupnosti. Pokud posouváme členy doleva, tak dostáváme posloupnosti $(a_1, a_2, a_3, \dots)$, $(a_2, a_3, a_4, \dots)$, $(a_3, a_4, a_5, \dots)$ a tak dále, rozmyslete si, že se toto dá zapsat vzorcem $\{a_{k+N}\}_{k=0}^{\infty}$, kde $N$ udává, o kolik jsme členy posunuli.

Trochu obtížnější je posun doprava. Myslíme tím posloupnosti $(0, a_0, a_1, a_2, \dots)$, $(0, 0, a_0, a_1, \dots)$, $(0, 0, 0, a_0, \dots)$ a tak dále. Nabízí se zápis $\{a_{k-N}\}_{k=0}^{\infty}$, ale má to podstatný zádrhel. Představme si posun od dva, naivní pokus by byl $\{a_{k-2}\}_{k=0}^{\infty}$, ale jak vypadá první člen této posloupnosti? Pro $k = 0$ dostáváme $a_{-2}$, kteréžto číslo vůbec neexistuje. Toto se řeší zavedením Heavisideovy funkce $H(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0, \end{cases}$ a pak uvažujeme posloupnost $\{a_{k-N} H(k-N)\}_{k=0}^{\infty}$. Pro indexy $k \geq N$ je $H(k-N) = 1$ a tudíž členy neovlivní, naopak pro $k < N$ je $H(k-N) = 0$ a to se bere tak, že první členy jsou automaticky nulové a na $a_{k-N}$ už se ani nedíváme, čímž se elegantně vyhneme problémům.

Teď se podíváme, jak se tyto operace odrazí na našem přiřazení. V zásadě jde jen o aplikaci Věty 9d.5 na naší situaci.

**Věta 10d.3.** (gramatika)
Nechť $\{a_k\}_{k=0}^{\infty} \in M$. Pak platí následující:
(i) Pro $c \in \mathbb{R}$ platí $T(\{c^k a_k\}_{k=0}^{\infty}) = T(\{a_k\}_{k=0}^{\infty})(cx)$;
(ii) Pro $N \in \mathbb{N}$ platí $T(\{a_{k+N}\}_{k=0}^{\infty}) = \frac{1}{x^N}\Big[T(\{a_k\}_{k=0}^{\infty}) - \sum\limits_{k=0}^{N-1} a_k x^k\Big]$;
(iii) Pro $N \in \mathbb{N}$ platí $T(\{a_{k-N}H(n-N)\}_{k=0}^{\infty}) = x^N T(\{a_k\}_{k=0}^{\infty})$;
(iv) $T(\{ka_k\}_{k=0}^{\infty}) = x[T(\{a_k\}_{k=0}^{\infty})]'$;
(v) $T(\{(k+1)a_k\}_{k=0}^{\infty}) = [x\,T(\{a_k\}_{k=0}^{\infty})]'$.

Než to dokážeme, ukážeme si tatáž pravidla v jiném zápise, který je pro mnoho lidí (ale ne všechny) uživatelsky přítulnější. Zahrneme i linearitu, ať to máme všechno pohromadě.

**Důsledek 10d.4.** (pravidla)
Nechť $\{a_k\}_{k=0}^{\infty}, \{b_k\}_{k=0}^{\infty} \in M$, označme $f(x) = T(\{a_k\})$ a $g(x) = T(\{b_k\})$. Necht $c \in \mathbb{R}$ a $N \in \mathbb{N}$. Pak platí následujucí.
(1) $(a_0, a_1, a_2, \dots) + (b_0, b_1, b_2, \dots) \leftrightarrow f(x) + g(x)$;
(2) $\{ca_k\} = (ca_0, ca_1, ca_2, \dots) \leftrightarrow cf(x)$;
(3) $\{c^k a_k\} = (c^0 a_0, c^1 a_1, c^2 a_2, \dots) \leftrightarrow f(cx)$;
(4) $(a_N, a_{N+1}, a_{N+2}, \dots) \leftrightarrow \frac{1}{x^N}\big[f(x) - a_0 - a_1 x - \dots - a_{N-1}x^{N-1}\big]$;
(5) $(0, 0, \dots, 0, a_0, a_1, a_2, \dots) \leftrightarrow x^N f(x)$;
(6) $\{ka_k\} = (0 \cdot a_0, 1 \cdot a_1, 2a_2, 3a_3, \dots) \leftrightarrow x[f(x)]'$;
(7) $\{(k+1)a_k\} = (1 \cdot a_0, 2a_1, 3a_2, 4a_3, \dots) \leftrightarrow [x\,f(x)]'$.

Všimněte si, že druhý vzoreček ve slovníku (Fakt 10d.2) vznikne z prvního aplikováním pravidla (7) z této věty.

Teď tato pravidla dokážeme, vždy si napíšeme řadu odpovídající upravené posloupnosti nalevo a pak z ní nějak zkusíme vyrobit řadu odpovídající původní posloupnosti. Použijeme značení $f(x)$ pro $T(\{a_k\})$, protože nám to ulehčí život. Pokud vám některé úpravy přijdou jako černá magie, zkuste si zvolit konkrétní hodnotu (třeba $N = 4$) a napsat si ty vzorečky v dlouhé formě namísto sum.

**Důkaz** (poučný)**:** (i): Posloupnosti $\{c^k a_k\}$ odpovídá řada

$$\sum_{k=0}^{\infty}(c^k a_k)x^k = \sum_{k=0}^{\infty} a_k(c^k x^k) = \sum_{k=0}^{\infty} a_k(cx)^k = f(cx).$$

(ii): Posloupnosti $\{a_{k+N}\} = (a_N, a_{N+1}, a_{N+2}, \dots)$ odpovídá řada

$$\begin{aligned}
\sum_{k=0}^{\infty} a_{k+N}x^k &= \sum_{k=0}^{\infty} a_{k+N}x^{k+N-N} = \sum_{k=0}^{\infty} a_{k+N}x^{k+N}\frac{1}{x^N} = \frac{1}{x^N}\sum_{k=0}^{\infty} a_{k+N}x^{k+N} \\
&= \left|\begin{matrix} m = k+N \\ k \geq 0 \Longrightarrow\ m \geq N \end{matrix}\right| = \frac{1}{x^N}\sum_{m=N}^{\infty} a_m x^m = \frac{1}{x^N}\Big[\sum_{m=N}^{\infty} a_m x^m + \sum_{m=0}^{N-1} a_m x^m - \sum_{m=0}^{N-1} a_m x^m\Big] \\
&= \frac{1}{x^N}\Big[\sum_{m=0}^{\infty} a_m x^m - \sum_{m=0}^{N-1} a_m x^m\Big] = \frac{1}{x^N}\Big[f(x) - \sum_{m=0}^{N-1} a_m x^m\Big].
\end{aligned}$$

(iii): Posloupnosti $\{a_{k-N}H(k-N)\} = (0, \dots, 0, a_0, a_1, a_2, \dots)$ odpovídá řada

$$\begin{aligned}
\sum_{k=N}^{\infty} a_{k-N}x^k &= \left|\begin{matrix} m = k-N \\ k = m+N \\ k \geq N \Longrightarrow\ m \geq 0 \end{matrix}\right| = \sum_{m=0}^{\infty} a_m x^{m+N} = \sum_{m=0}^{\infty} a_m x^m x^N \\
&= x^N \sum_{m=0}^{\infty} a_m x^m = x^N f(x).
\end{aligned}$$

(iv): Posloupnosti $\{ka_k\} = (0, a_1, 2a_2, 3a_3, \dots)$ odpovídá řada

$$\sum_{k=0}^{\infty} k a_k x^k = x \sum_{k=0}^{\infty} a_k k x^{k-1} = x \sum_{k=0}^{\infty} a_k [x^k]' = x \sum_{k=0}^{\infty} [a_k x^k]'$$
$$= x \Big[\sum_{k=0}^{\infty} a_k x^k\Big]' = x[f(x)]'.$$

(v): Posloupnosti $\{(k+1)a_k\} = (a_0, 2a_1, 3a_2, 4a_3, \dots)$ odpovídá řada

$$\sum_{k=0}^{\infty} (k+1) a_k x^k = \sum_{k=0}^{\infty} a_k (k+1) x^k = \sum_{k=0}^{\infty} a_k [x^{k+1}]' = \Big[\sum_{k=0}^{\infty} a_k x^{k+1}\Big]'$$
$$= \Big[x \sum_{k=0}^{\infty} a_k x^k\Big]' = [x\, f(x)]'.$$

□

Jsou dva možné přístupy k práci s těmito pravidly a generujícími funkcemi vůbec. Je možné ignorovat „vnitřnosti" a při transformování mechanicky aplikovat slovník a gramatiku, kterou se uživatel prostě naučí nazpaměť. Tento přístup funguje vcelku uspokojivě u rutinních příkladů a má malé nároky na přemýšlení takového uživatele. Selhává ovšem při setkání s příkladem, který nějak vybočuje, takových je samozřejmě spousta, možná většina.

Proto je perspektivnější také věci rozumět, chápat podstatu transformace i způsob, jak se k oněm pravidlům přichází. Jednak to dodá sebejistoty při práci s pravidly, druhak to představuje záchrannou kotvu pro případ, že by paměť nefungovala úplně spolehlivě, a hlavně je to jediný způsob, jak přistupovat k nerutinním příkladům.

Jako příklad použití gramatiky si dokážeme pár dalších zajímavých vzorečků, které se občas hodí.

**Fakt 10d.5.**
(i) $(1, -1, 1, -1, 1, \dots) \leftrightarrow \frac{1}{1+x}$;
(ii) $(0, 1, 1, 1, 1, \dots) \leftrightarrow \frac{x}{1-x}$;
(iii) $\{a^k\} \leftrightarrow \frac{1}{1-ax}$;
(iv) $\big\{-\frac{1}{a^{k+1}}\big\} \leftrightarrow \frac{1}{x-a}$.

U (iv) není jasné, proč by někdo chtěl pracovat s tak ošklivou posloupností, ale na ten vzoreček je nutné se podívat z opačné strany: Bude se nám silně hodit umět transformovat funkce typu $\frac{1}{x-a}$.

**Důkaz** (poučný)**:** (i) Použijeme pravidlo (3) s volbou $c = -1$ na vzorec (i) z Faktu 10d.2. Nebo počkáme na důkaz (iii) a pak tam použijeme $a = -1$.
(ii) Pravidlo (5) na vzorec (i) z Faktu 10d.2.
(iii) Pravidlo (3) na vzorec (i) z Faktu 10d.2.
(iv) $\frac{1}{x-a} = \frac{1}{a} \frac{1}{\frac{x}{a}-1} = -\frac{1}{a} \frac{1}{1-\frac{1}{a}x} \leftrightarrow -\frac{1}{a}\big\{\big(\frac{1}{a}\big)^k\big\}$ pomocí (iii), dál už se to snadno upraví.

□

Tím máme všechny nástroje pohromadě a je načase si ukázat, k čemu generující funkce mohou být.

**Příklad 10d.c:** Uvažujme rekurentní rovnici $a_{n+2} - a_n = 3 \cdot 2^n - 12$ pro $n \geq 0$ s počátečními podmínkami $a_0 = 1$, $a_1 = 0$.

Víme, že daná rovnice vlastně reprezentuje mnoho rovnic:

$$a_2 - a_0 = 3 \cdot 2^0 - 12$$
$$a_3 - a_1 = 3 \cdot 2^1 - 12$$
$$a_4 - a_2 = 3 \cdot 2^2 - 12$$
$$\vdots$$

Vlastně tedy porovnáváme nekonečně mnoho čísel neboli postupně členy jistých posloupností. Na pravé straně vidíme posloupnost $\{3 \cdot 2^n - 12\}_{n=0}^{\infty}$. Na levé straně bude lepší si to rozložit, odečítá se tam neznámá posloupnost $\{a_n\}_{n=0}^{\infty}$ a přičítá posunutá posloupnost $\{a_{n+2}\}_{n=0}^{\infty}$. Danou rovnici lze tedy interpretovat jako rovnost mezi posloupnostmi:

$$\{a_{n+2}\}_{n=0}^{\infty} - \{a_n\}_{n=0}^{\infty} = \{3 \cdot 2^n - 12\}_{n=0}^{\infty}.$$

Když se posloupnosti rovnají, musí se rovnat i jejich obrazy vzhledem k zobrazení $T$, tedy díky linearitě a dalším pravidlům máme

$$T[\{a_{n+2}\}_{n=0}^{\infty} - \{a_n\}_{n=0}^{\infty}] = T[\{3\cdot 2^n - 12\}_{n=0}^{\infty}]$$
$$T[\{a_{n+2}\}_{n=0}^{\infty}] - T[\{a_n\}_{n=0}^{\infty}] = T[\{3\cdot 2^n\}_{n=0}^{\infty}] - T[\{12\}_{n=0}^{\infty}]$$
$$\tfrac{1}{x^2}\big[T(\{a_n\}_{n=0}^{\infty}) - a_0 - a_1x\big] - T(\{a_n\}_{n=0}^{\infty}) = 3T(\{2^n\}_{n=0}^{\infty}) - 12T(\{1\}_{n=0}^{\infty}).$$

Bude se nám lépe pracovat, když si označíme $f(x) = T(\{a_n\})$. Díky počátečním podmínkám také umíme dosadit za $a_0$ a $a_1$, to je ale náhodička, ono to tak musí vyjít už z principu (počet těch členů v prvním vzorci je roven stupni rovnice, což se rovná počtu počátečních podmínek). Také umíme vyhodnotit výrazy napravo. Dostáváme

$$\tfrac{1}{x^2}[f(x) - 1 - 0] - f(x) = 3\frac{1}{1-2x} - 12\frac{1}{1-x}.$$

Tuto rovnici teď vyřešíme pro neznámou funkci $f(x)$.

$$\begin{aligned}
\tfrac{1}{x^2}[f(x) - 1] - f(x) &= 3\frac{1}{1-2x} - 12\frac{1}{1-x}\\
f(x) - 1 - x^2f(x) &= \frac{3x^2}{1-2x} - \frac{12x^2}{1-x}\\
(1-x^2)f(x) &= 1 + \frac{3x^2}{1-2x} - \frac{12x^2}{1-x}\\
f(x) &= \frac{1}{1-x^2} + \frac{3x^2}{(1-x^2)(1-2x)} - \frac{12x^2}{(1-x^2)(1-x)}.
\end{aligned}$$

Máme funkci, teď bychom ji potřebovali pomocí $T^{-1}$ převézt zpět na posloupnost. Výrazy na pravé straně ale nemáme ve slovníčku (ani rozšířeném). Pomůže algebra, dané výrazy lze rozložit na parciální zlomky. Detaily necháme do kursu analýzy, dostáváme

$$\begin{aligned}
f(x) &= \frac{1}{(1-x)(1+x)} + \frac{3x^2}{(1-x)(1+x)(1-2x)} - \frac{12x^2}{(1-x)^2(1+x)}\\
&= \Big(\frac{\frac12}{1-x} + \frac{\frac12}{1+x}\Big) + \Big(\frac{-\frac32}{1-x} + \frac{\frac12}{1+x} + \frac{1}{1-2x}\Big) - \Big(-\frac{9}{1-x} + \frac{6}{(1-x)^2} + \frac{3}{1+x}\Big)\\
&= \frac{8}{1-x} - \frac{6}{(1-x)^2} - \frac{2}{1+x} + \frac{1}{1-2x}.
\end{aligned}$$

Tento výraz již dokážeme převést na posloupnosti. Pomocí $T^{-1}$ neboli čtení zprava doleva u $\leftrightarrow$ dostáváme

$$\{a_n\}_{n=0}^{\infty} = 8 - 6(k+1) - 2\cdot(-1)^k + 2^k = 2^k - 2\cdot(-1)^k - 6k + 2.$$

Přesně tento výsledek jsme dostali, když jsme tento problém řešili jako příklad 10b.h pomocí algoritmu přes charakteristická čísla, přidruženou homogenní rovnici a podobně.

△

**Příklad 10d.d:** Najdeme obecné řešení rovnice $a_{n+2} - 3a_{n+1} + 2a_n = 0$.

Naše metoda potřebuje znalost počátečních podmínek, tak si tam dáme parametry $a_0 = p$, $a_1 = q$. Teď aplikujeme transformaci na rovnici, kterou vnímáme jako vztah o posloupnostech:

$$\{a_{n+2}\}_{n=0}^{\infty} - 3\{a_{n+1}\}_{n=0}^{\infty} + 2\{a_n\}_{n=0}^{\infty} = \{0\}_{n=0}^{\infty}.$$

Když si obraz $\{a_n\}_{n=0}^{\infty}$ označíme jako $f(x)$, dostáváme

$$\begin{aligned}
\frac{1}{x^2}[f(x) - a_0 - a_1x] - 3\frac{1}{x}[f(x) - a_0] + 2f(x) &= 0\\
[f(x) - p - qx] - 3x[f(x) - p] + 2x^2f(x) &= 0\\
f(x)[1 - 3x + 2x^2] &= p + qx - 3px\\
f(x) &= \frac{p + qx - 3px}{1 - 3x + 2x^2} = \frac{p + qx - 3px}{(2x-1)(x-1)}\\
f(x) &= \frac{q-2p}{x-1} + \frac{p-q}{2x-1} = (2p-q)\frac{1}{1-x} + (q-p)\frac{1}{1-2x}.
\end{aligned}$$

Zpětná transformace dává $\{a_n\}_{n=0}^{\infty} = \{(2p-q) + (q-p)\cdot 2^n\}_{n=0}^{\infty}$.

Není to úplně nejtradičnější tvar. Vzhledem k tomu, že $p, q$ jsou libovolné, můžeme si označit $v = q - p$, čímž dostáváme $a_n = (p - v) + v2^n$, a díky svobodě volby $p$ pak ještě máme i $p - v$ s libovolnou hodnotou, můžeme jej označit $u$. Dostáváme tak řešení $a_n = u + v2^n$, což opět odpovídá řešení, které bychom dostali klasickým

způsobem. U tohoto příkladu by se díky charakteristickým číslům získalo na jednom řádku.
△

Když porovnáme řešení Algoritmem 10b.8 a přes generující funkce, vidíme následující rozdíly:
• Máme-li najít obecné řešení, bývá klasická metoda přes vlastní čísla často výrazně rychlejší.
• Potřebujeme-li čistě partikulární řešení, pak je metoda pomocí transformace přinejmenším rovnocenná, může být i snažší.

Kritické body řešení přes generující funkce jsou dva. Jednak musíme být schopni vyřešit explicitně rovnici, kterou jsme odvodili pro funkci $f$, a pak ještě musíme umět tuto funkci transformovat zpět (parciální zlomky jsou možná pracné, ale zaručeně fungují, může být hůř). Výhodou této metody je, že je docela flexibilní. Na této úrovni to ovšem moc neuvidíme. Když si představíme, jaké pravé strany bychom uměli s našimi znalostmi transformovat, tak nám v zásadě zase vyjdou kvazipolynomy, na levé straně pak dostáváme rozumné rovnice v případech, kdy jde o lineární rekurentní rovnici. Jinými slovy, s tím, co zatím umíme, se transformací dají řešit v zásadě stejné rovnice jako pomocí metod z přecházejících sekcí. Potvrdí to i následující příklad.

**Příklad 10d.e:** Vyřešíme rovnici $a_{n+1}-(n+1)a_n=0$, $n\geq 0$ s počáteční podmínkou $a_0=1$.
Po transformaci pomocí $T$ a s označením $T(\{a_n\})=f(x)$ dostáváme

$$\begin{aligned}\frac{1}{x}[f(x)-1]-[xf(x)]'&=0\\ \frac{1}{x}[f(x)-1]-f(x)-xf'(x)&=0\\ f(x)-1-xf(x)-x^2f'(x)&=0\\ x^2f'(x)+(x-1)f(x)+1&=0\end{aligned}$$

Toto je dosti drsná diferenciální rovnice, takže se dál nedostaneme. Ani tato metoda nepomůže.
Zajímavé je, že transformace je užitečná v obou směrech. Již jsme mluvili o převodu rekurentní rovnice na rovnici s funkcemi, ale velice užitečný může být i převod nepříjemné diferenciální rovnice na rekurentní, jejíž řešení se pak dá třeba nějak odhadnout.
Zrovna náš příklad to ale neukáže. Pokud někdo potřebuje vyřešit diferenciální rovnici výše a převede si ji transformací na rekurentní rovnici, může relativně snadno uhodnout řešení $a_n=n!$. Bohužel, již jsme diskutovali, že tato posloupnost nepatří do $M$, tudíž ji pomocí $T$ neumíme převést zpět na funkci a řešení diferenciální rovnice tak nedostaneme.
△

Výhody metody vyniknou ve chvíli, kdy si uživatel rozšíří slovníček a zásobu rozličných transformačních triků, navíc lze upravit i koncept generující funkce, aby obsáhl více posloupností. Tato rozšíření jsou mimo rozsah této kapitoly, jejím cílem je čtenáře dovést k pochopení, jakým mechanismem transformace fungují, bude pak pro něj snažší pracovat i s jinými. Obecný mechanismus je vždy stejný: 1. Daná rovnice se přetransformuje do zcela jiného jazyka, bonusem bývá, pokud se nepříjemné operace díky pravidlům převedou na příjemnější. 2. Nová rovnice se vyřeší. 3. Získané řešení se zase převede zpět.
Na závěr ukážeme pár zajímavých rozšíření slovníčku pro naši transformaci.

**Fakt 10d.6.**
(i) Pro $n\in\mathbb{N}$ platí $\bigl\{\binom{n}{k}\bigr\}=\bigl(\binom{n}{0},\binom{n}{1},\binom{n}{2},\dots,\binom{n}{n},0,0,\dots\bigr)\leftrightarrow(1+x)^n$.
(ii) Pro $\alpha\in\mathbb{R}$ platí $\bigl\{\binom{\alpha}{k}\bigr\}\leftrightarrow(1+x)^\alpha$.
(iii) Pro $n\in\mathbb{N}$ platí $\bigl\{\binom{n+k}{n}\bigr\}=\bigl(1,n+1,\binom{n+2}{n},\binom{n+3}{n},\dots\bigr)\leftrightarrow\frac{1}{(1+x)^{n+1}}$.

Jak tyto vzorečky naznačují, generující funkce jsou užitečným nástrojem i v kombinatorice.

## Cvičení

**Cvičení 10d.1:** Najděte $T(\{k^2\})$.

**Řešení:**
**10d.1:** $\frac{x(x+1)}{(1-x)^3}$.