Teorie algoritmů

Natalie Žukovec

4. cvičení 2021

$f(n) \in \Theta(n^{\log_b a})$ $1 \notin \Theta(n^{\log_3 2})$

Funkce $T(n)$ je dána na přirozených číslech rekurentním vztahem. Vyřešte její asymptotické chování. Řešení zdůvodněte.

$T(n) = 2\,T(\frac{n}{3}) + 1$, $T(1) = 1$.

$a = 2$, $b = 3$ $\log_b a = \log_3 2 < 1$

$f(n) = \underbrace{1}_{n^0} \in O(n^{(\log_3 2) - \varepsilon})$, pro nějaké $\varepsilon > 0$

$\log_3 2 - \varepsilon \geq 0$ $0 < \varepsilon \leq \log_3 2$

$\boxed{\text{MT1}}$

$T(n) \in \Theta(n^{\log_3 2})$

Funkce $T(n)$ je dána na přirozených číslech rekurentním vztahem. Vyřešte její asymptotické chování. Řešení zdůvodněte.

$T(n) = 4\,T(\frac{n}{3}) + n^{\frac{3}{2}} \lg n$, $T(1) = 1$.

$a = 4 \quad b = 3$ $\quad \log_b a = \log_3 4 < \frac{3}{2} = \log_3 3^{\frac{3}{2}} = \log_3 \sqrt{27}$

$\sqrt{16}$

$f(n) = n^{\frac{3}{2}} \lg n \in \Omega(n^{\log_3 4 + \varepsilon})$

$\sqrt{16} < \sqrt{27}$

$\boxed{\text{MT3}}$ pro $0 < \varepsilon \leq \frac{3}{2} - \log_3 4$

$\varepsilon = \frac{3}{2} - \log_3 4 > 0$

$\log_3 4 + \varepsilon \leq \frac{3}{2}$

$\varepsilon \leq \frac{3}{2} - \log_3 4$

$a f(\frac{n}{b}) \leq c f(n)$, pro $c < 1$

$$4f\left(\frac{n}{3}\right) \le c\, f(n)$$

$$\underbrace{4\left(\frac{n}{3}\right)^{\frac{3}{2}} \lg \frac{n}{3}} \le c\, n^{\frac{3}{2}} \lg n$$

$$\frac{4}{\sqrt{27}} \cdot n^{\frac{3}{2}} \left(\lg n - \underbrace{\lg 3}_{>0}\right) \le \frac{4}{\sqrt{27}}\, n^{\frac{3}{2}} \lg n \quad , \quad \underbrace{c = \frac{4}{\sqrt{27}} < 1}$$

$$T(n) \in \Theta\left(n^{\frac{3}{2}} \lg n\right)$$

Funkce $T(n)$ je dána na přirozených číslech rekurentním vztahem. Vyřešte její asymptotické chování. Řešení zdůvodněte.

$T(n) = T(n-1) + c^n$, $T(1) = 1$, a kde $c > 1$ je konstanta z $\mathbb{R}^+$.

zostouá

$T(1) = 1$

$T(2) = 1 + c^2$

$T(3) = 1 + c^2 + c^3$

$\vdots$

$T(n) \in \Theta(c^n)$

$c^n \leq T(n) \leq \frac{c}{c-1} \cdot c^n$

>0

$+c$

$T(n) = 1 + c^2 + c^3 + \dots + c^n = \sum_{i=2}^{n} c^i + 1$

$= c^n \sum_{i=0}^{n-2} \frac{1}{c^i} + 1 \leq c^n \sum_{i=0}^{\infty} \frac{1}{c^i} = c^n \frac{1}{1 - \frac{1}{c}}$

$+c$

1) $c^n + c^{n-1} + \dots c^2 + 1$ $\quad T(n) \in \Theta(c^n)$ $\quad f(n) = c^n$

2) $n^c + (n-1)^c + \dots 2^c + 1^c$ $\quad T(n) \in \Theta(n \cdot n^c) = \Theta(n^{c+1})$

$\sum_{i=1}^{n} i^c \in \Theta(n f(n))$

$f(n) = n^c > 0$, rostoucí

$f(\frac{n}{2}) \in \Theta(f(n))$, pak ....

1) $T(n) = T(n-1) + c^n$, $T(1) = 1$, a kde $c > 1$ je konstanta z $\mathbb{R}^+$.

2) $T(n) = T(n-1) + n^c$, $T(1) = 1$, a kde $c \geq 1$ je konstanta z $\mathbb{R}^+$.

3) $T(n) = T(n-1) + 2$, $T(1) = 1$.

$$\lim_{n \to \infty} \frac{(n/2)^c}{n^c} = \left(\frac{1}{2}\right)^c$$

3) $2 + 2 + \dots + 2 + 1$

$2(n-1) + 1 \in \Theta(n)$

$m = \lg n$

$\boxed{n = 2^m}$, $\sqrt{n} = 2^{\frac{m}{2}}$

$T(n) = T(2^m) = S(m)$

$T(n) = T(\sqrt{n}) + 1$, $T(1) = 1$. (Použijte substituci $T(2^m) = S(m)$.)

$S(m) = S(\frac{m}{2}) + 1$

$a = 1$, $b = 2$, $\log_b a = \lg 1 = 0$

$f(m) = 1$, $m^0 = 1$

$f(m) \in \Theta(m^0)$

$MT2 \Rightarrow S(m) \in \Theta(\lg m)$

$T(n) \in \Theta(\lg(\lg n))$

$f(n) \in \Theta(n^{\log_b a} \cdot \lg^k n), \quad k \geq 0$ $\qquad n^{\log_b a} \lg^{k+1} n$

$n \cdot \lg n \quad (k=1) \qquad T(n) = \Theta(n \lg^2 n)$

Jde-li, vyřešte s použitím Master Theoremu následujíci vztah.

Nejde-li, zdůvodněte proč.

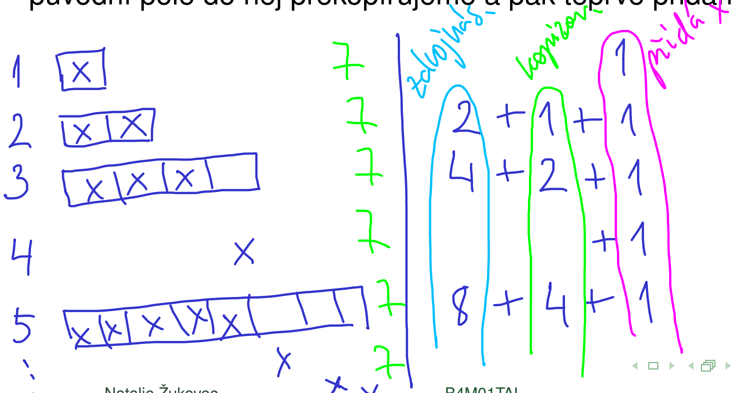
$T(n) = 2T(\frac{n}{2}) + n \log n$ $\qquad a=2, \; b=2 \quad \log_b a = \lg 2 = 1$

$f(n) = n \lg n \in \Omega(n^1)$ $\qquad$ ~~MT3~~

$n \lg n \notin \Omega(n^{1+\varepsilon})$, $n^{1+\varepsilon} = n \cdot n^{\varepsilon}$ $\qquad$ pro $\varepsilon > 0$ $\quad \varepsilon$ neex

$$\lim_{n\to\infty} \frac{n \lg n}{n^{1+\varepsilon}} = \lim_{n\to\infty} \frac{\lg n}{n^{\varepsilon}} = \left|\frac{\infty}{\infty}\right| \overset{L'H}{=} 0$$

Spočítejte amortizovanou složitost funkce $\boxed{pridej(x)}$, která do pole přidá prvek x. Funkce se realizuje takto: Začínáme z jednoprvkového pole, v okamžiku, kdy už je pole zaplňené, pole zdvojnásobíme, původní pole do něj překopírujeme a pak teprve přidáme x.



$$2^2 \le 5 < 2^3$$

$$T(n) = n + 3\left(2^{k+1} - 1\right) \le n + 3(2n-1) = 7n - 3 < 7n$$

$$\bullet + \bullet = 3\left(\underbrace{1 + 2 + 2^2 + 2^3 + \dots + 2^k}_{2^{k+1}-1}\right) = 3 \cdot \left(2^{k+1} - 1\right)$$

$$\underbrace{2^k \le n}_{2^{k+1} \le 2n} < 2^{k+1}$$

$$n < T(n) < 7n, \quad T(n) \in \Theta(n)$$

$$1 < \frac{T(n)}{n} < 7, \quad \frac{T(n)}{n} \in \Theta(1)$$

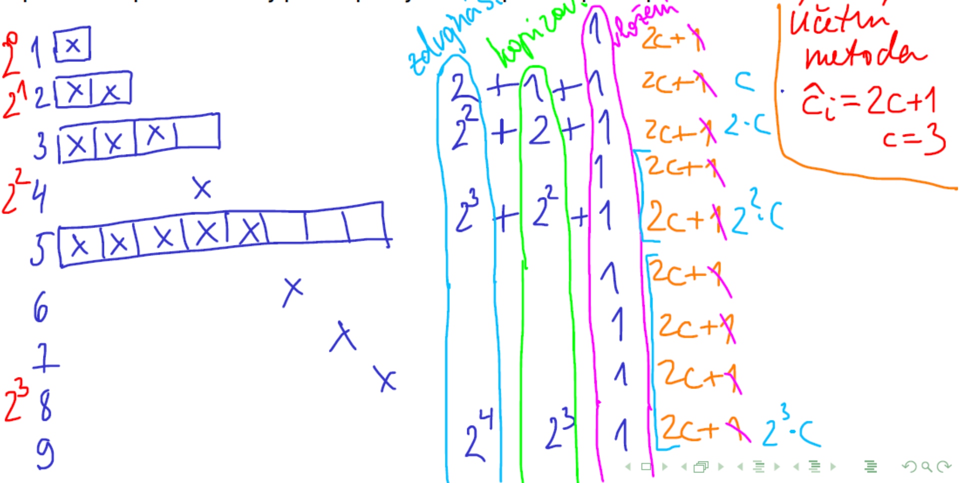
$$\hat{c} := 7 = 2c + 1$$

$$c = 3$$

Nejdelší čas jedné operace je $O(n)$ když $n = 2^k+1$

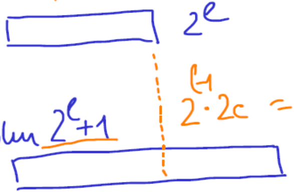
$O(1)$ $\quad T(n) = 2\cdot 2^k + 2^k + 1 = 3\cdot 2^k + 1 = 3n-2$

Spočítejte amortizovanou složitost funkce *pridej*(x), která do pole přidá prvek x. Funkce se realizuje takto: Začínáme z jednoprvkového pole, v okamžiku, kdy už je pole zaplňené, pole zdvojnásobíme, původní pole do něj překopírujeme a pak teprve přidáme x.



v kroku $2^{\ell-1}+1$

2^{ℓ}



$2^{\ell-1}\cdot 2c = 2^{\ell}\cdot c$ kreditů, které stačí na vytvoření pole délky $2^{\ell+1}$

v kroku $2^{\ell}+1$

$2\cdot 2^{\ell} = 2^{\ell+1}$

$$\underbrace{2^{\ell}+2^{\ell}}_{\text{zdroj.}} + \underbrace{2^{\ell}}_{\text{kopie}} = \underbrace{3}_{c}\cdot 2^{\ell}$$

$a, b \in \mathbb{N}$

Násobení dlouhých čísel: Mějme dvě binární čísla a, b a chceme spočítat jejich součin. Diskutujte časové složitosti a správnost následujících dvou algoritmů:

Algoritmus 1 (naivní): Předpokládejme, že součet dvou čísel trvá konstantní čas.

mezivysledek $:= 0$
  for $(i = 1; i < a + 1; i + +)$ do
    mezivysledek $:=$ mezivysledek $+ b$
return mezivysledek

$$\underbrace{b + b + b + \ldots + b}_{a}$$

$$O(a) = O(2^n)$$

má n cifer

$\underbrace{AC}_{1}, \underbrace{BD}_{2}, \underbrace{(A+B)(C+D)}_{3} = AC + AD + BC + BD$

Algoritmus 2 (rekurzivní): Libovolné $2N$-ciferné číslo můžeme zapsat jako $2^N A + B$, kde A a B jsou N-ciferná. Součin dvou takových čísel pak bude

$$(2^N \cdot A + B) \cdot (2^N \cdot C + D) = (2^{2N} \cdot \underbrace{AC}_{1} + 2^N (\underbrace{AD}_{2} + \underbrace{BC}_{5}) + \underbrace{BD}_{4}).$$

Předpokládejme, že sčítání proběhne v konstantním čase, stejně jako násobení mocninou 2. N-ciferná čísla budeme násobit rekurzivním zavoláním téhož algoritmu.

$c=5$

$T(n) = 4T\left(\frac{n}{2}\right) + cn$, c je konst

$a = 4$
$b = 2$
$\log_b a = \lg 4 = 2$

MT1

$n^{\lg 4} = n^2$

$f(n) = cn \in O(n^{2-\varepsilon})$, $0 < \varepsilon \leq 1$, $\varepsilon = \frac{1}{2}$

$T(n) \in \Theta(n^2)$

$T(n) = 3T\left(\frac{n}{2}\right) + cn$

$\boxed{k = \lg n \quad \frac{n}{2^k} = 1}$

$T(n) \in \Theta(n^{\lg 3})$

Určete, co bude rozhodujícím parametrem pro určení časové a paměťové složitosti, je-li na vstupu:

1. posloupnost prvků $a_1, a_2, \ldots a_n$, počet prvků n
2. graf o n vrcholech a m hranách, počet vrcholů a hran m, n
3. matice o rozměrech $n \times m$,
4. číslo x, jehož hodnota je podstatná pro celkovou délku výpočtu (například test prvočíselnosti).

1) hodnota x

2) počet cifer

Jsou dána dvě reálná čísla $a, b > 0$ a je dán pseudokód

while $a > 0$ do
  if $a < b$ then
    $(a, b) := (2a, b - a)$
  else
    $(a, b) = (a - b, 2b)$
end while
return b

$a = 1$ $b = 1$

0 2

$b = 2$

$a = 2$ $b = 1$

1 2

2 1

Jedná se o pseudokód algoritmu? Odpověď pečlivě zdůvodněte. Jestliže ano, spočítejte časové nároky algoritmu.

pro $b = 2a$

Určete asymptotickou složitost následující funkce v závislosti na vstupním parametru $N \in \mathbb{N}$, a to v nejhorším případě. V případě, že časová složitost není definována pro všechna N, určete pro která N definovaná je.

$i := N$
while $i > 0$ do
  $j := 0$
  while $j \cdot j < i$ do $j := j + 1$
  $i := i - 2$

$i = N \quad j \cdot j < N, \quad \sqrt{N}$

$i = N-2 \quad \sqrt{N-2}$

$i = N-4 \quad \sqrt{N-4}$

$\vdots$

$T(N) = \sqrt{N} + \sqrt{N-2} + \dots + \sqrt{2} =$

N je sudé $= \sum_{i=1}^{\frac{N}{2}} \sqrt{2i} = \sqrt{2} \sum_{i=1}^{\frac{N}{2}} \sqrt{i}$

$f(n) = \sqrt{n} >$ rostoucí

$f(\frac{n}{2}) \in \Theta(f(n))$

$\lim_{n \to \infty} \frac{\sqrt{\frac{n}{2}}}{\sqrt{n}} = \sqrt{\frac{1}{2}}$

$T(N) \in \Theta(N\sqrt{N})$

$\sum_{i=1}^{\frac{N}{2}} \sqrt{i} = \Theta(N\sqrt{N})$

N je liché

$T(N) = \sqrt{N} + \sqrt{N-1} + \dots + \sqrt{3} + \sqrt{1}$

$= \sum_{i=1}^{\frac{N+1}{2}} \sqrt{2i-1} \quad \in \Theta(N\sqrt{N})$

$N = 7$

$\left\lceil \frac{7}{2} \right\rceil$

$i := 7$ $j = 0$ $0 \cdot 0 < 7$

$j = 1$ $1 \cdot 1 < 7$

$j = 2$ $2 \cdot 2 < 7$

$j = 3$ $3 \cdot 3 > 7$

$\lceil \sqrt{7} \rceil + 1$

$i := 5$ $j = 0$ $0 \cdot 0 < 5$

$j = 1$ $1 \cdot 1 < 5$

$j = 2$ $2 \cdot 2 < 5$

$j = 3$ $3 \cdot 3 > 5$

$i := 3$ $j = 0$ $0 \cdot 0 < 3$

$j = 1$ $1 \cdot 1 < 3$

$j = 2$ $2 \cdot 2 > 3$

$i = 1$ $j = 0$ $0 \cdot 0 < 1$

$j = 1$ $1 \cdot 1 \geq 1$

$i = -1$

$N \quad N-2 \quad N-4 \quad N-6 \quad \dots \quad \begin{matrix} 0 \\ -1 \end{matrix}$

$$\left\lfloor \frac{N}{2} \right\rfloor \text{ pro} \begin{cases} N & \dots \text{ násobíme} & 1 + \lceil \sqrt{N} \rceil \\ N-2 & \text{—"—} & 1 + \lceil \sqrt{N-2} \rceil \\ \vdots & & \\ 2 & & \\ 1 & \text{—"—} & 1 + \lceil \sqrt{1} \rceil \quad (1 + \lceil \sqrt{2} \rceil) \end{cases}$$

Čím <u>omezíme</u> $T(N) \approx \left\lfloor \frac{N}{2} \right\rfloor + \sum_{i=0}^{\frac{N}{2}} \lceil \sqrt{N-2i} \rceil$

Jde o to mít asymptotický odhad, proto $T(N)$ nahradíme $\frac{N}{2} + \sum_{i=0}^{\frac{N}{2}} \sqrt{N-2i}$

K zjištění potřebujeme odhad pro

$$\sum_{i=0}^{\frac{N}{2}} \sqrt{N-2i} = \sum_{j=0}^{\frac{N}{2}} \sqrt{2j}$$

$f(n) = \sqrt{n}$ postoupíme $f\left(\frac{n}{2}\right) = \sqrt{\frac{n}{2}} = \frac{1}{\sqrt{2}} \cdot \sqrt{n} = \frac{1}{\sqrt{2}} f(n)$

ano, $f\left(\frac{n}{2}\right) \in \Theta(f(n))$

Proto $\sqrt{2} \sum_{i=1}^{\frac{N}{2}} \sqrt{i} \in \Theta\left(\frac{N}{2} \cdot \sqrt{\frac{N}{2}}\right) =$

$= \Theta\left(N^{\frac{3}{2}}\right) = \Theta\left(N \cdot \sqrt{N}\right)$

Víme $T(N)$ bylo asymp. $\boxed{\frac{N}{2} + \sqrt{2} \sum_{i=1}^{\frac{N}{2}} \sqrt{i}}$

Proto $\underline{T(N) \in \Theta\left(N \cdot \sqrt{N}\right)}$

$\left(\frac{N}{2} < N \cdot \sqrt{N} \right)$