Teorie algoritmů

Natalie Žukovec

5. cvičení 2021

$\cancel{\log_3 N \cdot \frac{N}{3} \in \Theta(N \lg N)}$ $T(n) \in \Theta(N)$

Určete asymptotickou složitost následující funkce v závislosti na vstupním parametru $N \in \mathbb{N}$, a to v nejhorším případě.

$i := N$
`while` $i > 0$ `do`
  $j := 0$
  $i := i/3$ celočíselně
  `while` $j < i$ `do` $j := j + 1$

$N = 9$

$i = 9$	$j = 0$,	$i = 3 = \frac{N}{3}$
$i = 3$	$j = 0$,	$i = 1 = \frac{N}{9}$
$i = 1$	$j = 0$	$i = 0$

$$T(N) = \underbrace{\boxed{\frac{N}{3}} + \frac{N}{3^2} + \ldots + \frac{N}{3^k}}$$

$\frac{N}{3^k} = 1$, $N = 3^k$
$k = \log_3 N$

$$= \sum_{m=1}^{k} \frac{N}{3^m} = N \sum_{m=1}^{k} \frac{1}{3^m} \leq N \sum_{m=1}^{\infty} \frac{1}{3^m} = N \frac{1/3}{1 - 1/3}$$

$q = \frac{1}{3}$ $\qquad \frac{N}{3} \leq T(n) \leq \frac{N}{2}$

$z = m \cdot n$

Jsou dána dvě kladná přirozená čísla m, n

$x := m; y := n; z := 0$

while $x \neq 0$ do

  if $x \pmod 2 = 1$ then

    $z := y + z$

  end if

  $x := \lfloor \frac{x}{2} \rfloor$

  $y := y \cdot 2$

end while

return z

variant

$m = 10$ $n = 3$

$x = 10$, $x = 5$, $y = 6$ $5 \cdot 6 + 0$

$z = 6$, $x = 2$, $y = 12$ $2 \cdot 12 + 6$

$x = 1$ $y = 24$ $1 \cdot 24 + 6$

$z = 30$ $x = 0$ $y = 48$ $0 \cdot 48 + 30$

$\overbrace{x, \frac{x}{2}, \frac{x}{4}, \dots \frac{x}{2^k}}^{k+1} = 1$ $m = 2^k$, $k = \lg m$

1. Spočítejte časové nároky algoritmu, tj najděte funkci f takovou, že časové nároky jsou ve třídě $\Theta(f)$. $f = \lg m$
2. Zjistěte jakou hodnotu má proměnná z na konci práce algoritmu.
3. Dokažte své tvrzení z bodu 2., tj. najděte invariant a dokažte ho.

Invariant: před i po každém provedení while cyklu platí $\boxed{xy + z = mn}$

1. Na začátku $x = m$, $y = n$, $z = 0$ $\quad xy + z = mn$

2. Platí-li pro x, y, z platí i po provedení while cyklu:

A) $x \pmod 2 = 1$, $\quad z_1 = y + z$, $x_1 = \frac{x-1}{2}$, $y_1 = 2y$

$$x_1 y_1 + z_1 = \frac{x-1}{\cancel{2}} \cdot \cancel{2}y + y + z = xy \cancel{- y} + \cancel{y} + z = xy + z = mn$$

B) $x \pmod 2 = 0$ $\quad z_1 = z$, $x_1 = \frac{x}{2}$, $y_1 = 2y$

$$x_1 y_1 + z_1 = \frac{x}{2} \cdot 2y + z = xy + z = mn$$

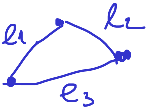
3. Na konci $x = 0$, $\quad z = mn$

Rozhodněte, zda je správný algoritmus pro nalezení minimální kostry.
1) V případě, že správný je, najděte variant a dokažte invariant.
2) V případě, že správný není, najděte protipříklad.

minKostra($G = (V, E)$, $a\colon E \to \mathbb{Z}$)
$T := \emptyset$
`foreach` $e \in E$ `do`
    `if` $T \cup \{e\}$ neobsahuje kružnici `then`
        $T := T \cup \{e\}$
`return` T

(V, T)

variant $m = |E|$



$a(e_1) = 3$
$a(e_2) = 3$
$a(e_3) = 1$

$T = \{e_1, e_2\}$ 6

min. kostra $\{e_1, e_3\}$ 4

Rozhodněte, zda je správný algoritmus pro nalezení minimální kostry.
V případě, že správný je, najděte variant a dokažte invariant.
V případě, že správný není, najděte protipříklad.

minKostra($G = (V, E)$, $a\colon E \to \mathbb{Z}$) variant $m = |E|$
$T := \emptyset$
`foreach` $e \in E$ `do`
    $T := T \cup \{e\}$
    `if` $T \cancel{\cup \{e\}}$ obsahuje kružnici C `then`
        $T := T \setminus \{e'\}$, kde e' je nejdražší hrana C
`return` T

Invariant: Po zpracování hran $L = \{e_1, e_2, \dots e_k\}$ je T min. les grafu (V, L)

1. Na začátku $L = \emptyset$, $T = \emptyset$ a tvrzení platí

2. Ať to platí pro L a vezmeme e_{k+1}

A) Pokud e_{k+1} neuzavře kružnici

$L' = L \cup \{e_{k+1}\}$, $\underline{T \cup \{e_{k+1}\}}$ je min les (V, L')

protože e_{k+1} je jediná hrana z S_1 do S_2,

B) Pokud e_{k+1} uzavře kružnici

Pak $S_3 \cup \{e_{k+1}\}$ má jedinou kružnici.

Když z ní odstraníme nejdražší hranu, dostaneme min strom S_3 v (V, L')

3. Na konci T je min $\underbrace{\text{souvislý les}}_{\text{strom}}$ (V, E)

min. kostra

algoritmus skončí ... hran je jen
konečně mnoho
a nejdelší hrana v kružnici
se najde v konečném čase

Invariant: E' = množina dosud
probraných hran

$T \quad (V, E')$
napnutý les (kostra se říkalo
napnutý strom)

Máme graf G s k komponentami.
Pak napnutý les je podgraf se stejnou
množinou vrcholů a stejnými komponentami,
když nemá kružnice (každá komponenta
je strom)

Před i po přidání hrany platí
T je minimální napnutý les
grafu (G, E').

- Platí na začátku
- Na konci dává, že T jsou hrany minimální kostry
- Když platilo před přidáním hran.
  - buď e spojovala 2 komp. souvislosti (V, T) – pak v E' je to jediná hrana mezi těmito komp.

→ nebo uzavřela kružnici
pak jsme odebrali nejdražší hranu
komp. zůstaly stejné a dostali
jsme min. napnutý les nového (V, E')