Zoltán Pálinkás
Zadaća 2

1.

$$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n))$$

$\boxed{\Leftarrow}$

$f(n) = \Theta(g(n)) \Rightarrow \exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N}$ t.d. je
$c_1 \cdot g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0$

Prema definiciji Θ notacije, $c_1 \cdot g(n) \leq f(n), \forall n \geq n_0$,
a to je zapravo definicija $\Omega(g(n))$.
$\Rightarrow f(n) = \Omega(g(n))$

Opet prema definiciji Θ notacije imamo $f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$,
a to je zapravo definicija $O(g(n))$
$\Rightarrow f(n) = O(g(n))$

$\boxed{\Rightarrow}$

Sada želimo pokazati da ako je $f(n) = O(g(n))$ i $f(n) = \Omega(g(n))$ onda je $f(n) = \Theta(g(n))$
Ako je $f(n) = O(g(n)) \Rightarrow \exists c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N}$ t.d. je $f(n) \leq c_2 \cdot g(n), \forall n \geq n_0$
Ako je $f(n) = \Omega(g(n)) \Rightarrow \exists c_1 \in \mathbb{R}^+, n_1 \in \mathbb{N}$ t.d. je $f(n) \geq c_1 \cdot g(n), \forall n \geq n_1$

Sada uzmimo max vrijednost n_0, n_1, $n_2 = \max\{n_0, n_1\}$

Sada imamo:

$c_1 \cdot g(n) \leq f(n), \forall n \geq n_2$
$c_2 \cdot g(n) \geq f(n), \forall n \geq n_2$
$\Rightarrow c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n), \forall n \geq n_2$

(2)

a) $T(n) = 16T\left(\frac{n}{4}\right) + n$

$a = 16$, $b = 4$, $f(n) = n$

$n^{\log_b a} = n^{\log_4 16} = n^2$

Slučaj 1. Ako $f(n) = O\left(n^{\log_b a - \varepsilon}\right)$ za neki $\varepsilon > 0$ tada $T(n) = \Theta\left(n^{\log_b a}\right)$

za $\varepsilon = 1 \Rightarrow f(n) = n = O\left(n^{2-1}\right) = O(n)$

$\Rightarrow T(n) = \Theta\left(n^2\right)$

b) $T(n) = 3T\left(\frac{n}{6}\right) + n^{0.500001}$

$a = 3$, $b = 6$, $f(n) = n^{0.500001}$

$n^{\log_b a} = n^{\log_6 3} = n^{0,61315}$

I ovdje vrijedi slučaj 1. $f(n) = n^{0.500001} = O\left(n^{0,61315 - \varepsilon}\right)$, za neki $\varepsilon > 0$

$\Rightarrow T(n) = \Theta\left(n^{0,61315}\right)$ (npr. $\varepsilon = 0,1$)

c) $T(n) = 3T\left(\frac{n}{4}\right) + n \log n$

$a = 3$, $b = 4$, $f(n) = n \log n$

$n^{\log_b a} = n^{\log_4 3} = n^{0,79248}$

Gledaj li moguće vrijediti slučaj 3.

1° $f(n) = n \log n = \Omega(n^{0.79248+\varepsilon})$, za neki $\varepsilon > 0$

vrijedi za npr. $\varepsilon \leq 0.2$

$\Rightarrow$ prvi dio je zadovoljen

2° $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$, za neki $c < 1$

$3 \cdot \frac{n}{4} \cdot \log\left(\frac{n}{4}\right) \leq c \cdot n \log n$ $\qquad \log \frac{n}{4} \leq \log n, \forall n \in \mathbb{N}$

$\frac{3}{4} \cdot \cancel{n} \cancel{\log n} \leq c \cdot \cancel{n \log n}$

$\frac{3}{4} \leq c \quad \Rightarrow c \in [\frac{3}{4}, 1\rangle$

$\Rightarrow$ drugi dio zadovoljen

$\Rightarrow T(n) = \Theta(n \log n)$

d) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

$a = 4, b = 2, f(n) = n^2$

$n^{\log_b a} = n^{\log_2 4} = n^2$

Slučaj 2.: $f(n) = n^2 = \Theta(n^2)$ ✓

$\Rightarrow T(n) = \Theta(n^2 \lg n)$

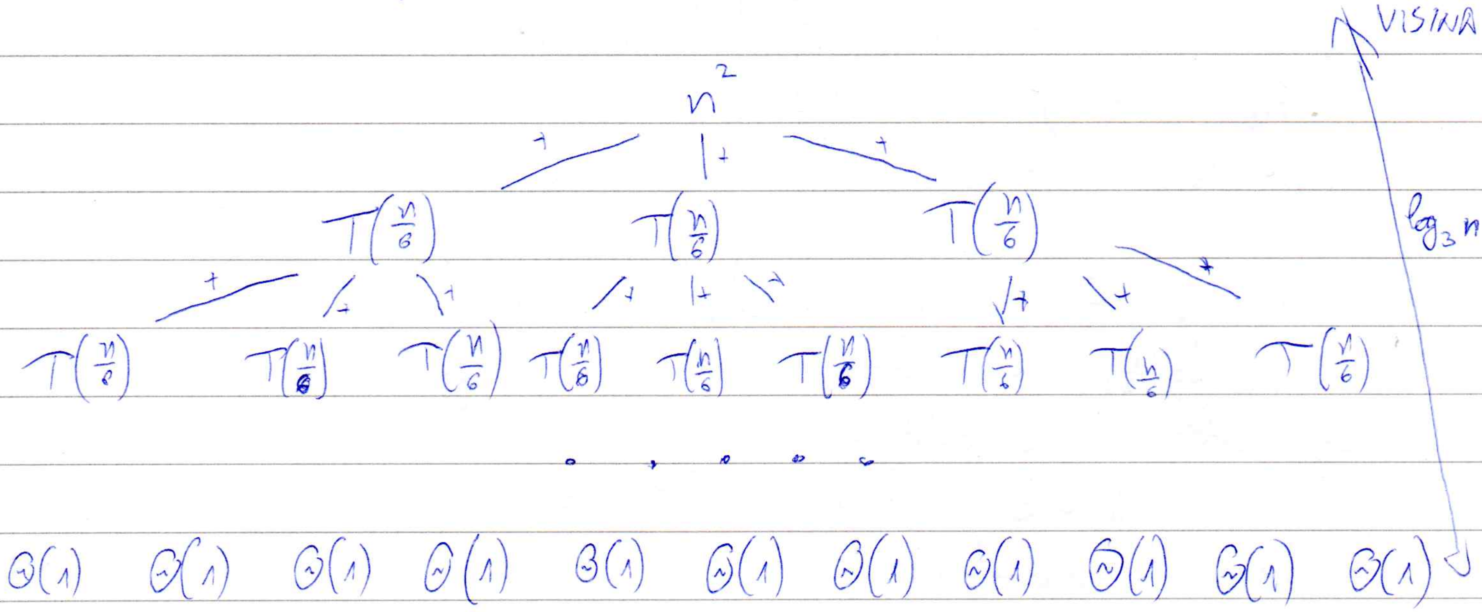
e) $T(n) = 4T\left(\frac{n}{2}\right) - n^2$

$a = 4, b = 2, f(n) = -n^2$

$\Rightarrow$ rješenje ove jednadžbe se ne može odrediti pomoću Master metode, jer je f-ja $f(n) = -n^2 < 0$, $\forall n \in \mathbb{N}$

3)

$$T(n) = 3T\left(\frac{n}{6}\right) + n^2$$



$$\begin{aligned} T(n) &= 3T\left(\frac{n}{6}\right) + n^2 \\ &= \log_3 n \cdot n^2 + n^2\, \Theta(1) \\ &= \Theta\left(n^2 \log_3 n\right) + \Theta\left(n^2\right) \\ &= \Theta\left(n^2 \log_3 n\right) \end{aligned}$$

4.

b) Vremenska složenost „Spoji" algoritma je $O(n+m)$, gdje je n duljina prve sortirane liste, a m duljina druge sortirane liste. Složenost je $n+m$, jer kroz obje liste prolazimo samo jednom i ubacujemo elemente na odgovarajuća mjesta.

c) $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$ ← sa predavanja

riješimo pomoću Masters metode

$a = 2, \quad b = 2, \quad f(n) = c \cdot n$

$n^{\log_b a} = n^{\log_2 2} = n$

$f(n) = cn = \Theta(n)$ ✓ (slučaj 2.)

$\Rightarrow T(n) = \Theta(n \lg n)$

d) Algoritam je „inplace", tj. sortira u polju u kojem su mu zadani podaci

$\Rightarrow \Theta(n)$ prostor

e) Rezultati za $n = 100$: Selection sort: ~53 μs

Merge sort: ~24 μs

$n = 1000$: Selection sort: ~5604 μs

Merge sort: ~2020 μs

$n = 1000$: Selection sort: ~471 ms

Merge sort: ~186 ms

$n = 1000000$: Selection sort:

Merge sort: No results.

$\Rightarrow$ Merge sort je puno brži od Selection sorta