

Kolokvij pri predmetu  
**ALGORITMI IN PODATKOVNE STRUKTURE 1**  
Visokošolski strokovni študij Računalništvo in informatika

točk

Ime in PRIIMEK: \_\_\_\_\_

Vpisna številka: \_\_\_\_\_ Podpis: \_\_\_\_\_

**Splošna navodila:** Natančno preberite navodila nalog. Odgovorite na zastavljena vprašanja. Odgovore utemeljite in obrazložite. **Pišite čitljivo.** Čas reševanja: 45 minut.

4 točk

**1. naloga:** Glede na veljavnost trditve v kvadrateg zapiši odgovor **da** ali **ne** (prazen kvadrat se šteje kot napačen odgovor):

- a)   $\sqrt{16} n^3 \log^4 n + 4n^3 = \Omega(n^2)$       b)   $3n^2 + 2n + 1 = \Omega(n^2 \log^{12} n)$   
c)   $2^{12} = O(\log^2 n)$       d)   $23n^2 \log n = O(n^2)$   
e)   $n^{\sqrt{\lg 16}} = \Theta(n^2)$       f)   $27^{\log_3 n} = \Theta(n^3)$   
g)   $\sum_{i=1}^n O(1) = O(1)$       h)   $\frac{(n+\lg n)^3}{n^3} = O(1)$

4 točk

**2. naloga:** Dana je časovna zahtevnost  $T(n) = 2n^2 + 3n + 4$  za nek algoritem.

- a) Koliko časa se bo algoritem izvajal pri nalogi velikosti 10.  
b) S pomočjo definicij asimptotične notacije pokaži, da velja  $T(n) = O(n^3)$ . Utemelji in jasno zapiši  $c$  in  $n_0$ .  
c) S pomočjo limit pokaži, da velja  $T(n) = \Theta(n^2)$   
d) Pokaži, da velja *tranzitivnost* za  $\Omega$  notacijo:  $f(n) = \Omega(g(n)) \wedge g(n) = \Omega(h(n)) \implies f(n) = \Omega(h(n))$ .

4 točk

**3. naloga:** Dano je *popolno dvojiško drevo* višine  $h$ . Vsakemu od listov tega drevesa pripnemo še (poljubno) poddrevo z  $i$  vozlišči, kjer je  $i$  zaporedna številka lista (liste številčimo od leve proti desni od 1 naprej).

- Kakšna je največja višina tako konstruiranega drevesa? Kakšna poddrevesa je potrebno pripeti v tem primeru?
- Koliko je vseh vozlišč v tako konstruiranem drevesu?
- Katero po vrsti je izpisano prvo vozlišče drugega pripetega poddrevesa, če naredimo premi obhod celotnega drevesa?
- Obrazloži razliko med *implicitno* in *eksplicitno* predstavitvijo dreves.

4 točk

**4. naloga:** Dan je sklad, ki vsebuje elemente (vrh sklada je na desni strani)

3, 1, 6, 2, 4, 7, 0, 8, 5

Elemente po vrsti jemljemo (operacija **pop**) s sklada in jih dodajamo v polje (po vrsti od indeksa 0 naprej). Nato iz polja zgradimo min-kopico (preko ugrezanja).

- Zapiši kopico, ki jo tako dobimo.
- Element 1 spremenimo v 5. Katero operacijo naredimo, da kopico popravimo?
- Zapiši algoritem, ki poišče drugi najmanjši element v kopici.
- Zapiši vmesni obhod drevesa (kopice).