

# OBLIKOVANJE I ANALIZA ALGORITAMA — popravni kolokvij

7. 2. 2011.

1. Neka su  $f, g : \mathbf{N} \rightarrow \mathbf{N}$  dvije funkcije. Napišite preciznu definiciju asimptotske  
(10) relacije ponašanja

$$f(n) \in O(g(n)) \quad (n \rightarrow \infty).$$

Koja svojstva ima relacija  $O$ ?

2. Između ponuđenih odgovora  
(10)

$$\Theta(1), \quad \Theta(\lg n), \quad \Theta(n), \quad \Theta(n \lg n), \quad \Theta(n^2), \quad \Theta(n^2 \lg n), \quad \Theta(n^3), \quad \Theta(2^n),$$

nađite točan red veličine za broj koliko puta se izvršava naredba  $x = x + 1$  u svakom od sljedećih dijelova programa ( $/$  je operator cjelobrojnog dijeljenja, kao u C-u):

```
(a) for i = 1 to n
    for j = 1 to i
        for k = 1 to j / 2
            x = x + 1;
```

```
(b) for i = 1 to n {
    j = 1;
    while (j <= n) {
        for k = 1 to j
            x = x + 1;
        j = j * 2;
    }
}
```

Ukratko **argumentirajte** odgovore!

3. Napišite algoritam koji računa **uniju**  $C = A \cup B$  dvaju skupova  $A$  i  $B$  koji sadrže  
(20) točno po  $n$  realnih brojeva. Skup brojeva prikazujemo uzlazno sortiranim poljem koje sadrži elemente tog skupa, uz pretpostavku da nema višestrukog pojavljivanja istih elemenata u polju. Algoritam treba vratiti broj  $k$  elemenata u skupu  $C$  i uzlazno sortirano polje njegovih elemenata (ako je  $k > 0$ , tj.  $C$  nije prazan).

Red veličine vremenske složenosti algoritma mora biti  $O(n \log n)$ . Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet.

Napomena: Broj bodova ovisi o složenosti algoritma. Složenost  $O(n \log n)$  vrijedi najviše 10 bodova. **Bonus:** složenost  $O(n)$  vrijedi 10 bodova više!

**OKRENITE!**

4. Zadan je skup od  $n$  poslova koje treba izvršiti. Izvršenje svakog posla traje jednu jedinicu vremena (na pr. 1 sekundu). U svakom trenutku  $t = 1, 2, \dots$ , možemo izvršiti točno **jedan** posao. Posao  $i$ , za  $1 \leq i \leq n$ , donosi profit  $p_i$ , ako i samo ako je njegovo izvršavanje započeto **najkasnije** do trenutka  $d_i$ . Treba naći redoslijed izvršavanja poslova koji maksimizira ukupni profit. Može se dogoditi da neke poslove nije moguće izvršiti u zadanom roku, tj. oni ne donose profit. Zbog toga treba naći broj  $k$  poslova čije izvršavanje donosi maksimalan profit i redoslijed  $i_1, i_2, \dots, i_k$  izvršavanja tih  $k$  poslova.

- (a) Nađite rješenje za  $n = 5$  poslova i sljedeće profite i rokove:

$i$	1	2	3	4	5
$p_i$	15	30	25	40	20
$d_i$	1	3	3	2	1

- (b) Sastavite algoritam koji nalazi optimalni redoslijed izvršavanja poslova. Treba vratiti broj  $k$  i polje  $red$ , duljine  $k$ , s tim da  $red[j] = i$  znači da posao  $i$  treba izvršiti kao  $j$ -ti po redu. Dokažite optimalnost algoritma.
- (c) Analizirajte složenost tog algoritma.

5. Zadano je polje od  $n$  objekata. Te objekte možemo uspoređivati uređajnom relacijom  $\leq$ . Definirajte što je problem sortiranja takvog polja i kako mjerimo složenost.

- (a) Opišite ukratko Quicksort algoritam za sortiranje polja.
- (b) Napišite **rekurziju** za složenost (ili neku mjeru složenosti) Quicksort algoritma. Što je rješenje te rekurzije u **najgorem** slučaju i kad se to događa, tj. kako tad izgleda polazno polje?
- (c) Ukratko opišite uz koje pretpostavke se izvodi **prosječna** složenost i koji rezultat se dobiva tom analizom.