

OBLIKOVANJE I ANALIZA ALGORITAMA – 1. kolokvij  
27.11.2013.

**Rezultati i uvid u kolokvije:** četvrtak, 5.12.2013., u 12:15

**Napomene:** Sva rješenja i pomoćne račune pišete isključivo na papirima koje će vam dati dežurni asistent/ica. Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje! Mobitele isključite i pospremite; nisu dozvoljeni niti kao zamjena za sat. Dozvoljeni načini zapisivanja algoritama u zadacima 3 i 4: bilo koji proceduralni jezik, uključivo i pseudokod. Argumenti funkcija/potprograma navode se s tipovima; petlje moraju biti korektno naznačene; poželjno je pisati komentare.

**Zadatak 1** (5+5) Odredite točan red veličine (relacija  $\Theta$ ) za broj izvršavanja naredbe  $x = x + 1$  u svakom od sljedećih dijelova programa (/ je operator cjelobrojnog dijeljenja, kao u C-u):

<pre>(a) i = 1;     while (i &lt;= n) {       for j = 1 to n         x = x + 1;       i = i * 2;     }</pre>	<pre>(b) i = n;     while (i &gt;= 1) {       for j = 1 to i         for k = 1 to i           x = x + 1;       i = i / 4;     }</pre>
--	---

Ukratko **argumentirajte** odgovore!

**Zadatak 2** (10) Zadana je rekurzivna relacija

$$T(n) = 4T(n/2) + f(n), \quad f(n) = n^2,$$

uz početni uvjet  $T(1) = d > 0$ . Nađite uvjetno asimptotsko ponašanje relacijom  $\Theta$  za rješenje  $T(n)$ , ako je  $n$  potencija od 2. Može li se dobiveno rješenje proširiti tako da asimptotsko ponašanje vrijedi bezuvjetno, za svaki dovoljno veliki  $n \in N$ , za rekurziju

$$T(n) = 2T(\lfloor n/2 \rfloor) + 2T(\lceil n/2 \rceil) + n^2, \quad \text{za } n \geq 2,$$

uz isti početni uvjet  $T(1) = d > 0$ ?

**Zadatak 3** (20) Neka je  $a$  uzlazno sortirano polje objekata nekog jednostavnog tipa. Treba provjeriti nalazi li se zadana vrijednost  $key$  u komadu polja  $a[i], \dots, a[j]$ . Ulazni argumenti algoritma su: polje  $a$ , indeksi  $i, j$ , i vrijednost  $key$ . Pretpostavljamo da zadani indeksi  $i, j$  korektno indeksiraju neke elemente u polju  $a$ , pa granice ne treba provjeravati, ali dozvoljavamo da je  $i > j$  na ulazu. Algoritam treba vratiti 1 (istina) ako postoji indeks  $k$  takav da je  $i \leq k \leq j$  i  $a[k] = key$ . U protivnom, treba vratiti 0 (laž).

- (a) Sastavite iterativni (**ne** rekurzivni) algoritam za **binarno** traženje odgovora, koji provjerava element na mjestu  $(i + j)/2$  i tu, po potrebi, “reže” polje. Kolika je njegova vremenska složenost u najgorem slučaju?

Zamislite da, u tom algoritmu, izraz  $(i + j)/2$  zamijenimo izrazom

- (b)  $i + (j - i)/3$ ,      (c)  $j - 2$ ,      (d)  $\max\{i, j - 2\}$ ,

bez ikakvih drugih promjena. Analizirajte pojedinačno svaki od tri dobivena algoritma. Radi li odgovarajući algoritam korektno, tj. nalazi li korektan odgovor na pitanje provjere? Ako da, kolika je njegova vremenska složenost u najgorem slučaju? Ako ne radi korektno, pokažite to primjerom i objasnite što se tada događa. Složenost svih algoritama mjerite brojem provjera.

**Zadatak 4** (20) Zadana su dva stringa  $x = x_1x_2 \dots x_n$  i  $y = y_1y_2 \dots y_m$ . Želimo naći duljinu njihovog najvećeg zajedničkog podstringa, tj. najveći  $k$  za koji postoje indeksi  $i$  i  $j$  takvi da vrijedi  $x_ix_{i+1} \dots x_{i+k-1} = y_jy_{j+1} \dots y_{j+k-1}$ . Koristeći dinamičko programiranje, sastavite algoritam koji na ulazu ima dva stringa i koji pronalazi traženi  $k$ .

Složenost algoritma mjerite brojem usporedbi znakova. Red veličine složenosti mora biti najviše  $O(mn)$ . Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet.