OBLIKOVANJE I ANALIZA ALGORITAMA — 1. kolokvij

23. 11. 2016.

1. (10) Nađite točan red veličine relacijom Θ za funkciju $incx(n)$ = broj koliko puta se izvršava naredba `x = x + 1` u svakom od sljedećih dijelova programa (`/` je operator cjelobrojnog dijeljenja, a funkcija `sqrt` vraća drugi korijen iz broja, kao u C-u):

(a)
```
i = 1;
while (i <= n) {
  for j = 1 to i
    for k = 1 to i
      x = x + 1;
  i = i * 3;
}
```

(b)
```
i = n;
while (i > 1) {
  j = i;
  while (j >= 1) {
    x = x + 1;
    j = j / 2;
  }
  i = (int) sqrt(i);
}
```

Ukratko **argumentirajte** odgovore!

2. (10) Zadana je rekurzivna relacija

$$T(n) = 3\,T(n/2) + f(n), \quad f(n) = n\sqrt{n},$$

uz početni uvjet $T(1) = d > 0$. Nađite uvjetno asimptotsko ponašanje relacijom Θ 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 $n \in \mathbb{N}$, za rekurziju

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

uz isti početni uvjet $T(1) = d > 0$?

OKRENITE!

3. (20) Zadani su prirodni broj n i cjelobrojno polje A s n elemenata, $A = (a_1, \ldots, a_n)$. Za svaki element a_i, tj. za $i = 1, \ldots, n$, u podnizu elemenata $(a_1, \ldots, a_{i-1})$ ispred a_i, treba naći indeks p_i zadnjeg (najdesnijeg) elementa koji je strogo veći od a_i. Ako u tom podnizu nema većeg elementa od a_i, ili ako je podniz prazan (na pr., za $i = 1$), po definiciji stavljamo da je $p_i = 0$. Formalno, treba naći polje indeksa p s n elemenata, tako da vrijedi

$$p_i = \max\{j \mid 0 \le j < i \text{ i vrijedi } a_j > a_i\}, \quad i = 1, \ldots, n,$$

uz dogovor $p_i = 0$ ako je navedeni skup prazan (ili možete uzeti da je $a_0 = \infty$ u gornjoj definiciji).

Napišite algoritam koji nalazi i vraća takvo polje indeksa p. Primjer za polje A s $n = 10$ elemenata:

i	1	2	3	4	5	6	7	8	9	10
a_i	3	2	5	4	1	2	3	7	3	6
p_i	0	1	0	3	4	4	4	0	8	8

Sve operacije na elementima i indeksima smatramo elementarnim operacijama, tako da složenost algoritma ovisi samo o n. U najgorem slučaju, red veličine vremenske složenosti algoritma mora biti $O(n^2)$. Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet.

Napomena: Broj bodova ovisi o složenosti algoritma u najgorem slučaju. Složenost $\omega(n)$ vrijedi najviše 10 bodova. **Bonus**: složenost $O(n)$, zajedno s analizom koja to pokazuje, vrijedi 10 bodova više!

4. (20) Zadana je cjelobrojna matrica A, reda n, u kojoj su elementi poredani tako da je svaki redak uzlazno sortiran i svaki stupac uzlazno sortiran, tj. vrijedi

$$\begin{aligned} a[i][j-1] &\le a[i][j], \quad j = 2, \ldots, n, \quad i = 1, \ldots, n, \\ a[i-1][j] &\le a[i][j], \quad i = 2, \ldots, n, \quad j = 1, \ldots, n. \end{aligned}$$

Napišite algoritam koji na ulazu dobiva tako sortiranu matricu A, reda n, i cijeli broj key. Algoritam treba vratiti `TRUE` ako postoji par indeksa (i, j) takav da je $a[i][j] = key$. U protivnom, treba vratiti `FALSE`. Smijete koristiti cjelobrojni prikaz logičkih vrijednosti `TRUE` $= 1$, `FALSE` $= 0$ (kao u C-u).

Složenost algoritma mjerimo točnim brojem usporedbi cijelih brojeva — elemenata matrice i zadanog broja key, u funkciji od n. U najgorem slučaju, algoritam smije imati **najviše** $4n$ usporedbi. Dokažite da vaš algoritam zadovoljava ovaj uvjet.

REZULTATI: nedjelja, 27. 11. 2016., navečer (na webu), ponedjeljak, 28. 11. 2016. u 12 sati.

Saša Singer