OBLIKOVANJE I ANALIZA ALGORITAMA — 1. kolokvij

19. 11. 2008.

1. (10) Između ponuđenih odgovora

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

nađite točan red veličine za broj koliko puta se izvršava naredba `x = x + 1` u svakom od sljedećih dijelova programa

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

(b)
```
j = n;
while (j >= 1) {
  for i = 1 to j
    x = x + 1;
  j = j / 3;
}
```

Ukratko argumentirajte odgovore.

2. (10) Zadana je rekurzivna relacija

$$T(n) = 7\,T(n/4) + f(n), \quad f(n) = n,$$

uz početni uvjet $T(1) = d > 0$. Nađite uvjetno asimptotsko ponašanje relacijom Θ za rješenje $T(n)$, ako je n potencija od 4. Može li se dobiveno rješenje proširiti tako da asimptotsko ponašanje vrijedi bezuvjetno, za svaki dovoljno veliki $n \in \mathbf{N}$, ako u rekurziji piše $\lceil n/4 \rceil$, umjesto $n/4$?

3. (20) Zadano je polje T od n elemenata koje možemo međusobno uspoređivati relacijom $\leq$. Treba naći najmanji i najveći element u polju T. Ako najmanjih ili najvećih elemenata ima nekoliko, dovoljno je naći bilo koji. Elementarna operacija je uspoređivanje jednog para elemenata, s rezultatom logičkog tipa (laž, istina). Mjera složenosti je **točan broj** takvih uspoređivanja u ovisnosti o n.

(a) Sastavite algoritam `minmax_1` kao kombinaciju standardnih algoritama za nalaženje najmanjeg, odnosno, najvećeg elementa i nađite njegovu složenost.

(b) Sastavite algoritam `minmax_2` za rješenje ovog problema koji se rekurzivno primjenjuje na "polovinama" ulaznog polja. Algoritam treba raditi za svaki n, bez obzira na parnost. Uz pretpostavku da je n potencija broja 2, nađite točno složenost ovog algoritma. Kako treba postupiti kod računanja složenosti za bilo koji n (rekurzija, kratka argumentacija, rješenje)?

(c) Obratite pažnju na "dno" rekurzije algoritma `minmax_2`. Može li se pažljivom realizacijom tog dijela algoritma dobiti **bolja** složenost nego za `minmax_1`?

Oba algoritma trebaju imati što manju složenost!

(Bodovi: (a) + (b) = 10, (c) = 10.)

OKRENITE!

4. (10) Napišite algoritam koji računa **presjek** $C = A \cap B$ dvaju skupova A i B koji sadrže 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!

REZULTATI: petak, 21. 11. 2008. u 14 sati. Saša Singer