OBLIKOVANJE I ANALIZA ALGORITAMA — 1. kolokvij

7. 11. 2012.

1. (10) Između ponuđenih odgovora

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

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

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

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

Ukratko **argumentirajte** odgovore!

2. (10) Zadana je rekurzivna relacija

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

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 $n \in \mathbf{N}$, za rekurziju

$$T(n) = 2\,T(\lfloor n/4 \rfloor) + 2\,T(\lceil n/4 \rceil) + n \lg n, \quad \text{za } n \geq 2,$$

uz početne uvjete $T(0) = 0$ i $T(1) = d > 0$?

3. (20) Zadano je polje $A[1..n]$ od n cijelih brojeva. Brojevi u tom polju smiju biti bilo kojeg predznaka (kao u tipu `int`). Svakom potpolju $A[i..j]$, gdje je $1 \leq i \leq j \leq n$, pridružujemo sumu svih elemenata u tom potpolju

$$S(i, j) = \sum_{k=i}^{j} A[k].$$

Sastavite algoritam koji nalazi onaj par indeksa i^*, j^*, za koji je ova suma najveća. Ako takvih parova ima više, dovoljno je pronaći samo jedan par (bilo koji od njih). Algoritam treba vratiti nađeni par indeksa i^*, j^* i pripadnu najveću sumu $S(i^*, j^*)$.

Zabranjeno je koristiti dodatna polja! Složenost algoritma mjerimo brojem zbrajanja cijelih brojeva. Red veličine složenosti algoritma mora biti najviše $O(n^2)$. Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet.

Napomena: Broj bodova ovisi o složenosti algoritma. Složenost $O(n^2)$ vrijedi najviše 10 bodova. **Bonus**: složenost $O(n \log n)$, zajedno s analizom koja to pokazuje, vrijedi 10 bodova više! Usput, postoji i algoritam složenosti samo $\Theta(n)$.

OKRENITE!

4. (20) U običnom problemu Hanojskih tornjeva imamo 3 štapa, jedan pored drugog, i n diskova, međusobno različitih veličina. U početnom položaju, svi diskovi se nalaze na prvom štapu, poredani po veličini, tako da je najveći na dnu, a najmanji na vrhu. Cilj je prebaciti sve diskove na treći štap (u isti poredak), a srednji štap služi kao pomoćni. U svakom koraku (jednom potezu) smijemo prebaciti samo **jedan** disk s nekog štapa na neki (bilo koji) drugi, s tim da je **zabranjeno** staviti veći disk iznad nekog manjeg diska.

U "ograničenom" problemu Hanojskih tornjeva imamo još jedno **dodatno** ograničenje: u svakom potezu, disk se smije prebaciti s nekog štapa, samo na **susjedni** štap, tj. nije dozvoljeno izravno prebacivanje diska s prvog na treći štap, ili, obratno, s trećeg na prvi. Drugim riječima, u svakom potezu, ili stavljamo disk na srednji štap, ili mičemo disk sa srednjeg štapa.

(a) Sastavite algoritam koji rješava ovaj "ograničeni" problem Hanojskih tornjeva u **najmanjem** mogućem broju poteza.

(b) Nađite pripadni točan **broj** poteza u ovisnosti o n (a ne samo red veličine).

(c) Dokažite da je to najmanji mogući broj poteza, tj. da je algoritam zaista optimalan.

U realizaciji algoritma smijete koristiti funkciju

```
prebaci( int odakle, int kamo )
```

koja realizira jedan potez prebacivanja (najgornjeg) diska sa štapa numeriranog brojem `odakle`, na štap s brojem `kamo`. Složenost algoritma mjerimo brojem poziva te funkcije.

Uputa: Algoritam je rekurzivan i vrlo sličan algoritmu za obični problem Hanojskih tornjeva. Probajte riješiti problem za $n = 1$ i $n = 2$, a onda pogledajte kako biste n-ti disk prebacili s prvog na treći štap (bez obzira na to koliko još diskova, eventualno, ima ispod n-tog na prvom štapu). Za dokaz optimalnosti pokažite (indukcijom) da bilo koji drugi algoritam mora imati barem toliko poteza kao i ovaj.

REZULTATI: petak, 9. 11. 2012. u 13 sati. Saša Singer