# OBLIKOVANJE I ANALIZA ALGORITAMA — 1. kolokvij

## 18. 11. 2015.

1. (10) Nađite točan red veličine relacijom $\Theta$ za funkciju $incx(n)$ = 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 {
    j = 1;
    while (j <= i) {
      x = x + 1;
      j = j * 2;
    }
}
```

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

Ukratko **argumentirajte** odgovore!

2. (10) Zadana je rekurzivna relacija

$$T(n) = 2\,T(n/3) + f(n), \quad f(n) = \log_3 n,$$

uz početni uvjet $T(1) = d > 0$. Nađite uvjetno asimptotsko ponašanje relacijom $\Theta$ za rješenje $T(n)$, ako je $n$ potencija od 3. Može li se dobiveno rješenje proširiti tako da asimptotsko ponašanje vrijedi bezuvjetno, za svaki dovoljno veliki $n \in \mathbf{N}$, za rekurziju

$$T(n) = T(\lfloor n/3 \rfloor) + T(\lceil n/3 \rceil) + \log_3 n, \quad \text{za } n \geq 2,$$

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

3. (20) Konačni skup $S = \{a_1, \ldots, a_n\}$, koji sadrži točno $n$ realnih brojeva, zadajemo brojem $n$ i strogo uzlazno sortiranim poljem $a$ svih njegovih elemenata.

Zadana su dva skupa $A$ i $B$, oba s točno po $n$ elemenata. Napišite algoritam koji računa **simetričnu razliku** $C = A \,\Delta\, B = (A \setminus B) \cup (B \setminus A)$. Algoritam treba vratiti broj $k$ elemenata u skupu $C$ i strogo 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 $O(n)$, zajedno s analizom koja to pokazuje, vrijedi 10 bodova više!

**OKRENITE!**

4. (20) Promatramo problem "sortiranja palačinki": pred konobarom je stog od $n$ palačinki, koje su sve međusobno različitih veličina i nalaze se jedna na drugoj (slično kao u Hanojskim tornjevima). Konobar ima lopaticu kojom može **naopačke** izokrenuti "blok" od **najgornjih** $k$ palačinki, tako da njihov međusobni poredak postaje obratan od prethodnog. Broj $k$ okrenutih palačinki može biti između 2 i $n$ (okretanje jedne palačinke ništa ne mijenja, jer ne razlikujemo gornju i donju stranu palačinke). Zadatak konobara je poredati palačinke od najmanje (na vrhu) do najveće (na dnu), tj. sortirati ih uzlazno po veličini, od vrha prema dnu.

(a) Sastavite algoritam koji realizira ovakvo sortiranje polja palačinki. Za zapis algoritma, početni položaj palačinki (od vrha prema dnu) opisan je poljem $a$ s $n$ elemenata, koje sadrži promjere palačinki — cijele brojeve, za koje pretpostavljamo da su svi međusobno različiti (ne treba provjeravati). Operacija "flip" naopakog okretanja najgornjih $k$ palačinki zadana je funkcijom

$$\mathit{flip}(a, k)$$

koja izokreće poredak **prvih** $k$ elemenata u polju $a$.

Za početak, složenost algoritma mjerimo samo brojem okreta, tj. brojem poziva funkcije *flip*, neovisno o broju $k$ okrenutih palačinki. U najgorem slučaju, algoritam smije imati **najviše** $2n$ okreta. Dokažite da vaš algoritam zadovoljava ovaj uvjet.

(b) Napišite efikasnu realizaciju funkcije *flip*, koja ima što je moguće manji broj **zamjena** parova elemenata u polju $a$, za zadani broj $k$.

Pretpostavimo da su jedna **usporedba** para elemenata i jedna **zamjena** para elemenata u polju $a$, elementarne operacije. Koliko je elementarnih operacija svake vrste potrebno, u najgorem slučaju, za sortiranje palačinki ovim algoritmom?

**REZULTATI:** srijeda, 2. 12. 2015. u 14 sati. Saša Singer