

# OBLIKOVANJE I ANALIZA ALGORITAMA — popravni kolokvij

13. 2. 2019.

1. Napišite precizne definicije sljedećih pojmova:
- (10) (a) za funkcije  $f, g : \mathbb{N} \rightarrow \mathbb{N}$  vrijedi  $f(n) \in \Omega(g(n))$ , kad  $n \rightarrow \infty$ ,  
(b) funkcija  $f$  **polinomno raste**.

Neka su  $f, g : \mathbb{N} \rightarrow \mathbb{N} \setminus \{1\}$  rastuće funkcije. Dana je sljedeća tvrdnja:

$$f(n) \in O(g(n)) \implies \lg f(n) \in O(\lg g(n)).$$

Ako je ova tvrdnja istinita, dokažite ju. U protivnom, nađite kontraprimjer i detaljno ga opravdajte.

2. Zadana je rekurzivna funkcija za ispis crtica (znakova '-'):

(15) 

```
void Crtice(int n) {
    int m = n / 2, i;
    if (n > 1) {
        Crtice(m); Crtice(m); Crtice(n - m);
        for (i = 1; i <= m; ++i) printf("-");
    }
    return;
}
```

Izračunajte točan broj ispisanih crtica za  $n = 8$  i  $n = 9$ . Nađite točan red veličine (relacijom  $\sim$ ) za broj ispisanih crtica, u funkciji od  $n$ , uz pretpostavku da je  $n \geq 0$ .

3. Zločesti kralj ima podrum s  $n$  bačvi vina. Špijun je uspio ubaciti otrov u jednu bačvu, ali kralj ne zna u koju. Otrov je vrlo djelotvoran, ali spor — bez obzira na to koliko je otrov razrjeđen, **bilo koja** količina otrova je dovoljna da ubije neku osobu i osoba umire nakon točno 30 dana od konzumacije otrova. Kralj je spreman žrtvovati neki broj svojih robova kao “kušače” vina, ali želi što prije saznati koja je bačva otrovana. Na kraljevu žalost, broj robova je bitno manji od broja bačvi u podrumu.

Nađite postupak kojim kralj, nakon točno 30 dana (tj. u najkraće moguće vrijeme), može odrediti koja od  $n$  bačvi je otrovana, s tim da žrtvuje **najviše**  $O(\log n)$  robova. Opis postupka smije biti kombinacija pseudo-kôda i običnih rečenica, ali mora biti dovoljno precizan. Argumentirajte da taj postupak zaista pronalazi otrovanu bačvu u zadanom roku, uz zadano ograničenje na broj robova.

Može li se problem riješiti žrtvujući **manje** robova, ali uz **dulji** rok? Primjerice, je li moguće iskoristiti samo jednog roba? Ako je, koliki je onda rok u kojem kralj nalazi otrovanu bačvu?

**OKRENITE!**

4. Na Otoku ljubavi nalazi se  $n$  zaljubljenih osoba. Svakoj od njih treba poželjeti sretno  
 (30) Valentinovo. Tome služi razglas na sredini otoka, kojeg mogu čuti sve osobe, ali samo ako su budne. Znamo da je osoba  $i$  budna samo u jednom vremenskom intervalu  $[a_i, b_i]$ , s tim da je  $a_i < b_i$ , dok u ostala vremena spava i ne može čuti razglas. Plan je preko razglasa reći “Sretno Valentinovo” u neka izabrana vremena  $t_1, \dots, t_m$ . Sve osobe koje su budne u nekom od tih vremena  $t_j$  čut će tu poruku. Vremena  $t_1, \dots, t_m$  treba izabrati tako da s najmanjim **brojem** poruka (**najmanjim**  $m$ ) osiguramo da svaka osoba čuje poruku bar jednom. Dozvoljeno je da ista osoba čuje poruku više puta. Ulaz algoritma su broj  $n$  i polje intervala  $[a_i, b_i]$ , za  $i = 1, \dots, n$ , a izlaz su (minimalni) broj  $m$  i polje izabranih vremena  $t_j$ , za  $j = 1, \dots, m$ .
- (a) Nađite optimalnu strategiju za nalaženje (izbor) vremena  $t_1, \dots, t_m$ , s najmanjim brojem  $m$  izabranih vremena za poruke. Riječima precizno opišite tu strategiju i dokažite njezinu optimalnost.
- (b) Sastavite pripadni algoritam i analizirajte njegovu vremensku složenost u ovisnosti o  $n$ . Složenost algoritma mora biti u  $O(n^2)$ .
5. Zadano je polje od  $n$  elemenata. Te elemente možemo uspoređivati **binarnom** uređajnom relacijom  $\leq$ . Definirajte što je problem sortiranja takvog polja i kako mjerimo složenost.
- (a) Kolika je **donja** ograda složenosti za **bilo koji** algoritam sortiranja polja koji koristi samo binarnu operaciju uspoređivanja elemenata u polju? Ukratko argumentirajte kako dolazimo do te ograde.
- (b) Skicirajte Quicksort algoritam za sortiranje polja i ukratko opišite osnovne dijelove tog algoritma.
- (c) Napišite **rekurziju** za složenost (ili neku mjeru složenosti) Quicksort algoritma. Što je rješenje te rekurzije u **najgorem** slučaju i kad se to događa, tj. kako tad izgleda polazno polje? Što je rješenje te rekurzije u **najboljem** slučaju? Može li Quicksort, u najboljem slučaju, imati linearnu složenost  $O(n)$ ? Ako da, kako tad izgleda polazno polje?
- (d) Ukratko opišite uz koje pretpostavke se izvodi **prosječna** složenost Quicksort algoritma i koji rezultat se dobiva (ne treba dokazivati rezultat).