OBLIKOVANJE I ANALIZA ALGORITAMA — 1. kolokvij

20. 11. 2019.

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, kao u C-u):

(a)
```
for i = 1 to n {
  j = n / i;
  while (j > 0) {
    x = x + 1;
    j = j - 1;
  }
}
```

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

Ukratko, ali dovoljno precizno, argumentirajte svaki odgovor!

2. (10) Zadana je rekurzivna funkcija za ispis znakova `'+'` (`/` je cjelobrojno dijeljenje):

```
void Plus(int n) {
    int i;
    if (n > 1) {
        Plus(n / 3);  Plus((n + 1) / 3);  Plus((n + 2) / 3);
    }
    for (i = 1; i <= n; ++i) printf("+");
    return;
}
```

Neka je $T(n)$ točan broj ispisanih znakova za zadani $n \geq 0$. Izračunajte $T(9)$ i $T(10)$. Nađite uvjetno asimptotsko ponašanje relacijom Θ za $T(n)$, ako je n potencija od 3. Može li se dobiveno rješenje proširiti tako da asimptotsko ponašanje vrijedi bezuvjetno, za svaki dovoljno veliki $n \in \mathbb{N}$?

OKRENITE!

3. (20) Treba naći lokalni maksimum (tzv. "**vrh**") u polju s n elemenata, odnosno, u matrici s $n \times n$ elemenata, gdje je n zadani prirodni broj. Elementi su (primjerice) cijeli brojevi. Dodatno, pretpostavljamo da su svi elementi u polju/matrici međusobno **različiti**. Precizne formulacije problema su u nastavku zadatka. Složenost algoritma mjerimo brojem usporedbi elemenata u polju/matrici. U oba slučaja treba pokazati da algoritam radi korektno i da zadovoljava traženu složenost.

(a) U polju A s n elemenata, element $A[i]$ je **vrh**, ako je strogo veći od oba susjedna elementa $A[i-1]$ i $A[i+1]$. Ako je $A[i]$ na rubu polja, dovoljno je da je strogo veći od onog susjeda koji se nalazi unutar polja, uz $n \geq 2$.

Sastavite algoritam koji nalazi neki vrh u polju A i vraća njegov indeks i. Složenost algoritma **mora** biti u $O(\log n)$.

(b) U matrici A s $n \times n$ elemenata, element $A[i][j]$ je **vrh**, ako je strogo veći od sva četiri susjedna elementa $A[i-1][j]$, $A[i+1][j]$, $A[i][j-1]$, $A[i][j+1]$ (dijagonalne susjede ne gledamo). Ako je $A[i][j]$ na rubu ili u kutu matrice, dovoljno je da je strogo veći od svih susjeda koji se nalaze unutar matrice (tri ili dva susjeda, uz $n \geq 2$).

Sastavite algoritam koji nalazi neki vrh u matrici A i vraća njegove indekse i, j. Složenost algoritma **mora** biti u $O(n \log n)$.

Može li se pretpostavka o različitosti **svih** elemenata oslabiti, a da navedeni algoritmi još uvijek rade korektno?

4. (20) 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.

(a) 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.

Konkretno pitanje: Ako kralj ima 1000 bačvi vina, je li 10 robova dovoljno da odredi otrovanu bačvu nakon točno 30 dana? Ako da, kako?

(b) 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?

Konkretno pitanje: Ako kralj ima 1000 bačvi vina, a slavlje je planirano za 5 tjedana, je li 8 robova dovoljno da odredi otrovanu bačvu prije tog roka? Ako da, kako?

REZULTATI: nedjelja, 24. 11. 2019., navečer (na webu),
ponedjeljak, 25. 11. 2019. u 12 sati.

Saša Singer