OBLIKOVANJE I ANALIZA ALGORITAMA — popravni kolokvij

12. 2. 2020.

1. (10) Napišite precizne definicije sljedećih pojmova:

   (a) za funkcije f, $g : \mathbb{N} \to \mathbb{N}$ vrijedi $f(n) \in o(g(n))$, kad $n \to \infty$,

   (b) funkcija f **nadeksponencijalno** raste. Navedite primjer takve funkcije f i pokažite da f nadeksponencijalno raste.

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

   $$f(n) \in \Omega(g(n)) \implies 3^{f(n)} \in \Omega(3^{g(n)}).$$

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

2. (15) Zadana je rekurzivna funkcija za ispis znakova `'0'` (`/` je cjelobrojno dijeljenje):

```
void Nula(int n) {
    int i;
    if (n > 1) {
        Nula(n / 3);  Nula((n + 2) / 3);
    }
    for (i = 1; i <= n; ++i) printf("0");
    return;
}
```

   Neka je $T(n)$ točan broj ispisanih znakova za zadani $n \geq 0$. Izračunajte $T(8)$ i $T(9)$. 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 $n \in \mathbb{N}$?

3. (30) Zadana je **rastuća** funkcija $f : \mathbb{N} \to \mathbb{Z}$, koju možete pozvati kao funkciju f, s jednim argumentom $i \in \mathbb{N}$, a funkcija vraća vrijednost $f(i)$. Treba naći **najmanji** argument $n \in \mathbb{N}$, za kojeg je $f(n) > 0$, tj. $n = \min\{i \in \mathbb{N} \mid f(i) > 0\}$, ako takav argument n postoji. Složenost algoritma mjerimo **brojem** poziva funkcije f.

   (a) Pretpostavljamo da je funkcija f **strogo** rastuća, tj. da je $f(i) < f(i+1)$, za svaki $i \in \mathbb{N}$. Pokažite da traženi n sigurno postoji. Sastavite što efikasniji algoritam za nalaženje n.

   (b) Pretpostavljamo da je funkcija f rastuća, ali ne mora biti strogo rastuća, već samo znamo da **postoji** $n_0 \in \mathbb{N}$ (kojeg **ne znamo**), takav da je $f(n_0) > 0$. Sastavite što efikasniji algoritam za nalaženje traženog n.

   U oba slučaja, složenost algoritma **mora** biti u $O(\log n)$. Dajte kratku (ali preciznu) argumentaciju da je složenost algoritma zaista u $O(\log n)$.

OKRENITE!

4. (30) Softverska tvrtka treba kupiti licence za razne komunikacijske protokole. Zbog važećih propisa, tvrtka može kupiti najviše jednu licencu mjesečno. Svaka licenca trenutno (na početku cijele kupovine) košta c_0 eura. Međutim, svaki sljedeći mjesec, svaka licenca postaje sve skuplja: cijena licence j raste za faktor $r_j > 1$. To znači da licenca j, kupljena nakon m mjeseci od početka, košta $c_0 \cdot r_j^m$ eura. Tvrtka mora kupiti n licenci i treba naći plan kupovine koji **minimizira** ukupnu cijenu, s tim da su zadani faktori rasta cijene $r_1, \ldots, r_n$ za svaku licencu. Pretpostavljamo da su svi faktori rasta međusobno **različiti**, tj. vrijedi $r_i \neq r_j$, za $i \neq j$.

(a) Nađite redoslijed u kojem treba kupovati licence, tako da ukupna cijena bude **minimalna**. Dokažite optimalnost tog redoslijeda. Uputa: analizirajte slučaj $n = 2$.

(b) Sastavite algoritam koji nalazi optimalni redoslijed kupovine i nađite njegovu složenost. Ulazni argumenti su brojevi n, c_0 i polje faktora rasta r, a izlaz je permutacija brojeva od 1 do n, koja sadrži optimalni redosljed kupovine, i pripadna minimalna ukupna cijena C. Složenost algoritma mora biti u $O(n^2)$. Ukratko argumentirajte da vaš algoritam zadovoljava taj uvjet.

5. (35) Vrlo velike prirodne brojeve $u \in \mathbb{N}$ prikazujemo nizom (poljem) znamenki u nekoj fiksnoj bazi $b \geq 2$. Zadana su dva takva n-znamenkasta broja u i v.

(a) Neka je $r \in \mathbb{N}$ unaprijed zadan i neka je n djeljiv s $r + 1$. Opišite rekurzivni algoritam za brzo množenje brojeva u i v, koji koristi strategiju "podijeli pa vladaj" na $r + 1$ blokova znamenki. Treba navesti ideju algoritma i skicirati 4 osnovna koraka.

(b) Kolika je vremenska složenost tog algoritma u ovisnosti o n i r? Dovoljno je navesti rekurziju za složenost i asimptotsku gornju ogradu za rješenje. Za zadani realni broj $\alpha > 1$, kako treba izabrati r, tako da dobijemo algoritam $M_\alpha(u, v)$ koji množi dva n-znamenkasta broja u i v u vremenu $T_\alpha(n) \in O(n^\alpha)$?

(c) Zadan je slijedeći algoritam za množenje velikih prirodnih brojeva u i v, gdje je n_0 neka unaprijed zadana konstanta, neovisna o u i v.

function *SuperMul* (u, v) {
    dopuni manjeg od brojeva u i v nulama sprijeda, ako treba,
        tako da oba broja imaju točno po n znamenki;
    if $n \leq n_0$ **then**
        pomnoži u i v standardnim algoritmom M_2 ;
    else {
        $\alpha := 1 + (\lg \lg n)/ \lg n$;
        pomnoži u i v algoritmom M_α ;
    }
    return izračunati produkt ;
}

Množi li *SuperMul* n-znamenkaste brojeve u vremenu $T(n) \in O(n \log n)$? Precizno obrazložite!

REZULTATI: četvrtak, 13. 2. 2020., navečer (na webu),
petak, 14. 2. 2020. u 12 sati.

Saša Singer