OBLIKOVANJE I ANALIZA ALGORITAMA — popravni kolokvij

10. 2. 2016.

1. (10) Napišite precizne definicije sljedećih pojmova:

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

   (b) funkcija f **eksponencijalno raste**.

   Kojoj klasi rasta pripada funkcija $f : D \to f(D)$

$$f(x) = 2^{\sqrt{x}},$$

   gdje je D prirodna domena ove funkcije? Precizno argumentirajte odgovor.

2. (10) Zadana je rekurzivna relacija

$$T(n) = 4\,T(n/3) + f(n), \quad f(n) = n^2,$$

   uz početni uvjet $T(1) = d > 0$. Nađite uvjetno asimptotsko ponašanje relacijom Θ 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

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

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

3. (35) Investicijska tvrtka planira buduća ulaganja na osnovu analize ponašanja dionica na tržištu u prethodnom razdoblju. U postupku analize često treba riješiti zadaće sljedećeg oblika. Promatra se ponašanje cijene određene dionice u bloku od n uzastopnih dana u prošlosti. Za svaki pojedini dan $i = 1, \ldots, n$, poznata je cijena c_i jedne dionice za taj dan (pretpostavimo da je cijena c_i konstantna kroz cijeli dan i).

   Uzmimo da tvrtka u tom periodu želi **kupiti** 100 dionica u nekom danu k, a zatim **prodati** sve te dionice u nekom **kasnijem** danu p. Na osnovu poznatih podataka treba naći koji dan je trebalo kupiti dionice i kada ih je trebalo prodati da zarada tvrtke bude **najveća** moguća. Ako u tom periodu od n dana **nije** bilo moguće ostvariti ikakvu zaradu, onda, umjesto traženih dana k i p, treba vratiti $k = p = 0$.

   Na primjer, neka je $n = 4$, a poznate cijene jedne dionice kroz ta četiri dana su $c_1 = 7$, $c_2 = 2$, $c_3 = 4$ i $c_4 = 5$ novčanih jedinica. Onda treba vratiti $k = 2$ (kupi na dan 2) i $p = 4$ (prodaj na dan 4), jer to daje zaradu od 3 novčane jedinice po svakoj dionici.

   Sastavite algoritam koji, za zadani n i polje cijena c, nalazi optimalne dane k i p, ako oni postoje, odnosno, vraća $k = p = 0$, ako zarada nije moguća.

   Red veličine složenosti algoritma mora biti $O(n \log n)$. Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet. Uputa: iskoristite princip "podijeli, pa vladaj" na polovinama zadanog vremenskog perioda (do $n/2$ dana i nakon toga).

OKRENITE!

4. (30) Tvrtka za sigurnost treba kupiti licence za razne dijelove kriptografskih programa. Zbog važećih propisa, tvrtka može kupiti najviše jednu licencu mjesečno. Svaka licenca trenutno (na početku cijele kupovine) košta 100 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 t mjeseci od početka, košta $100 \cdot r_j^t$ 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 broj n i polje faktora rasta r, a izlaz je permutacija brojeva od 1 do n, koja sadrži optimalni redosljed kupovine, i pripadna minimalna cijena c. Složenost algoritma mora biti $O(n^2)$.

5. (35) Zadano je polje od n objekata. Te objekte možemo uspređivati **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) Ukratko opišite Quicksort algoritam za sortiranje polja.

(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 i može li Quicksort u najboljem slučaju imati linearnu složenost $O(n)$?

(d) Ukratko opišite uz koje pretpostavke se izvodi **prosječna** složenost i koji se rezultat dobiva tom analizom (ne treba dokazivati rezultat).

REZULTATI: petak, 12. 2. 2016. u 12 sati. Saša Singer