OBLIKOVANJE I ANALIZA ALGORITAMA – 2. kolokvij

5.2.2014.

Rezultati i uvid u kolokvije: utorak, 11.2.2014., 9:00

Napomene: Sva rješenja i pomoćne račune pišete isključivo na papirima koje će vam dati dežurni asistent/ica. Dozvoljeno je korištenje isključivo pribora za pisanje i brisanje! Mobitele isključite i pospremite; nisu dozvoljeni niti kao zamjena za sat. Dozvoljeni načini zapisivanja algoritama: bilo koji proceduralni jezik, uključivo i pseudokod. Argumenti funkcija/potprograma navode se s tipovima; petlje moraju biti korektno naznačene; poželjno je pisati komentare.

Zadatak 1 (20) Natjecanje u trijatlonu ima tri dijela: plivanje u malom bazenu na 800 m, zatim vožnju biciklom na 10 km i, na kraju, trčanje na 3 km. Za svakog pojedinog natjecatelja nema odmora između ovih dijelova. Zbog sigurnosnih propisa, u bazenu se u bilo kojem trenutku smije nalaziti samo **jedan** natjecatelj. Na biciklističkoj i trkačoj stazi nema takvog ograničenja, tj. više natjecatelja može istovremeno biti na svakoj od tih staza.

Umjesto skupnog starta, natjecatelji startaju jedan za drugim — čim prethodni natjecatelj izađe iz bazena, tog trena starta sljedeći natjecatelj, i tako redom, pa se bazen koristi bez prekida.

Ukupno je prijavljeno n natjecatelja, a prijava i-tog natjecatelja sadrži očekivana vremena za svaki od tri dijela utrke: p_i za plivanje, b_i za vožnju bicikla i t_i za trčanje.

Za minimizaciju troškova pratnje, organizatori natjecanja žele naći **redosljed** kojim trebaju startati natjecatelji tako da **cijela** utrka — od starta prvog natjecatelja, do trena kad svi natjecatelji završe trčanje, ima **najkraće** očekivano trajanje (izračunato na osnovu očekivanih vremena iz prijava).

(a) Analizirajte slučaj $n = 2$ i nađite **pohlepni** kriterij koji daje najmanje trajanje cijele utrke. Dokažite optimalnost tog kriterija za bilo koji n.

(b) Sastavite algoritam koji nalazi optimalni startni redosljed i nađite njegovu složenost. Ulazni argumenti su broj n i polja očekivanih vremena p, b i t. Izlaz algoritma je permutacija brojeva od 1 do n, koja sadrži optimalni startni redosljed, i pripadno najmanje očekivano trajanje cijele utrke T. Složenost algoritma mora biti $O(n^2)$.

Zadatak 2 (25) Zadano je potpuno binarno stablo T s n čvorova, gdje je $n = 2^d - 1$, za neki $d \geq 1$. Svaki čvor v tog stabla označen je nekim realnim brojem x_v. Možete pretpostaviti da su sve ove oznake međusobno **različiti** realni brojevi. Kažemo da je čvor v **lokalni minimum** u T, ako je njegova oznaka x_v manja od oznake x_u, za sve čvorove u s kojima je v izravno pozvezan bridom (granom) u T. Uočite da takvih susjednih čvorova u može biti najviše 3 (roditelj, lijevo i desno dijete od v). Treba naći neki (bilo koji) lokalni minimum od T.

(a) Pokažite da svako potpuno binarno stablo T ima lokalni minimum. U trivijalnom binarnom stablu ($n = 1$), korijen je, po definiciji, lokalni minimum.

Oznake čvorova nisu posebno popisane pa im ne možemo izravno pristupati, već su "ugrađene" u sam čvor. To znači da, za bilo koji čvor v, oznaku x_v možemo očitati samo tako da "ispitamo" ili posjetimo taj čvor, po pravilima obilaska binarnog stabla — obilazak počinje u nekom čvoru, a pomak iz čvora dozvoljen je samo u neki susjedni čvor.

Možete uzeti da je svaki čvor stabla opisan strukturom tipa `node`, koja sadrži oznaku čvora x_v i dvije adrese, `left` i `right`, lijevog i desnog djeteta. Listovi stabla prepoznaju se po posebnoj adresi `NULL`, kao signal da nemaju djece. Po potrebi, možete u strukturu `node` dodati i treću adresu `parent`, za roditelja svakog čvora (korijen se onda prepoznaje po `NULL` adresi roditelja).

(b) Pretpostavimo da obilazak stabla T počinje u **korijenu** stabla, zadanom kao ulazni argument `root`. Sastavite algoritam koji nalazi i vraća neki lokalni minimum tog binarnog stabla. Složenost algoritma mjerimo brojem "ispitivanja" (ili posjeta) čvorova. Početni čvor se, također, broji. Red veličine složenosti algoritma mora biti $O(n)$ u najgorem slučaju. Analizirajte složenost vašeg algoritma i pokažite da ona zadovoljava ovaj uvjet. Napomena i uputa: pripazite na to da algoritam radi za mala stabla, kad je $d = 1, 2, 3$.

(c) Uzmimo da obilazak može početi u **bilo kojem** čvoru stabla T, a ne samo u korijenu, s tim da u svakom čvoru pamtimo i adresu roditelja. Početni čvor zadaje se kao ulazni argument algoritma. Može li se algoritam iz (b) jednostavno modificirati tako da radi i u ovom slučaju, a da za složenost i dalje vrijedi ista ocjena u najgorem slučaju? Ako da, pokažite kako. U protivnom, argumentirajte primjerom što se može dogoditi.

Napomena: Broj bodova ovisi o složenosti algoritma. Složenost $O(n)$ vrijedi najviše 10 bodova. **Bonus**: složenost $O(\log n)$ vrijedi 10 bodova više!

Zadatak 3 (20) Zadana su dva polinoma A i B s kompleksnim koeficijentima.

(a) Opišite osnovne korake **brzog** algoritma za računanje produkta $C = A \cdot B$ zadanih polinoma, korištenjem brze diskretne Fourierove transformacije (FFT) vektora čija duljina je potencija broja 2. Kolika je složenost tog algoritma?

(b) Ukratko opišite što radi diskretna Fourierova transformacija vektora duljine $n = 2^k$. Skicirajte **iterativni** algoritam za **brzu** diskretnu Fourierovu transformaciju (FFT) i izvedite njegovu složenost, uz pretpostavku sekvencijalnog izvršavanja operacija.

(c) Ukratko opišite kako se dobiva algoritam za inverznu Fourierovu transformaciju.

Napomena: odgovori na pojedine dijelove zadatka vrednuju se nezavisno (5+10+5), tj. ako ne znate odgovoriti na (a) dio, smijete odgovoriti na ostale dijelove.