OBLIKOVANJE I ANALIZA ALGORITAMA — 2. kolokvij

31. 1. 2018.

1. (25) Uz dugačku ravnu cestu nalazi se n relativno rijetko poredanih kuća. Položaj i-te kuće zadan je koordinatom x_i, izraženom u kilometrima, obzirom na neku ishodišnu točku na toj cesti. Možete pretpostaviti je ulazno polje x (tipa `double`) već uzlazno sortirano, tj. da vrijedi $x_1 < \cdots < x_n$.

Telefonska kompanija želi postaviti antene za mobilnu telefoniju na određene točke duž te ceste i to tako da je svaka kuća udaljena najviše 4 kilometra od neke antene — u protivnom, ako je kuća izvan tog intervala, signal u kući je preslab.

Sastavite algoritam koji nalazi **najmanji** potrebni broj antena i njihove pozicije, tako da su sve kuće "pokrivene" dovoljno jakim signalom. Algoritam treba vratiti broj antena k i polje y koje sadrži koordinate antena $y_1, \ldots, y_k$ u uzlaznom poretku. Zabranjeno je korištenje dodatnih polja.

(a) Pokažite primjerom da pohlepna strategija "uzmi interval zadane duljine koji pokriva **najviše** (preostalih) nepokrivenih kuća" ne mora biti optimalna.

(b) Nađite pohlepnu strategiju koja uvijek nalazi pokrivanje s **najmanjim** brojem intervala (antena) k. Dokažite da ta strategija zaista nalazi rješenje s **najmanjim** brojem antena. Nađeno rješenje za koordinate antena y_j ne mora biti jedinstveno, ali mora zadovoljavati uvjet "pokrivanja" svih kuća.

(c) Sastavite efikasni algoritam za nalaženje najmanjeg k i (nekog) polja pozicija $y_1, \ldots, y_k$ za antene. Svi dijelovi algoritma moraju biti potpuno razrađeni, do razine osnovnih naredbi. Vremenska složenost algoritma mora biti u $O(n)$, tj. linearna u n. Opravdajte da vaš algoritam zadovoljava taj uvjet.

(d) Snaga signala opada s povećanjem udaljenosti od antene. Zato da izbjegne prigovore korisnika, kompanija želi "lokalno" optimizirati nađene pozicije y_j. Za dani j, gledaju se samo kuće koje su originalno pokrivene antenom na poziciji y_j. Treba naći poziciju z_j, tako da najslabiji signal u tim kućama ima najveću moguću snagu (maksimizacija minimuma). Skicirajte algoritam vremenske složenosti $O(n)$ za ovu lokalnu optimizaciju.

OKRENITE!

2. (25) Zamislite da sortiramo polja A s n elemenata, s tim da su zamjene elemenata u A vrlo skupe. Umjesto toga, koristimo samo zamjene indeksa u pomoćnoj permutaciji p, koju smo inicijalizirali na identitetu, tj. $p[i] = i$, za $i = 1, \ldots, n$. Na kraju sortiranja, u polju p dobivamo permutaciju za koju vrijedi

$$A[\,p[1]\,] \leq A[\,p[2]\,] \leq \cdots \leq A[\,p[n]\,].$$

Sastavite algoritam `preuredi`, kojemu su ulaz polje A s n elemenata i permutacija p za koju vrijedi gornji niz nejednakosti. Algoritam treba preurediti polje A u uzlazno sortirani poredak, tako da vrijedi

$$A[\,1\,] \leq A[\,2\,] \leq \cdots \leq A[\,n\,].$$

Zabranjeno je korištenje dodatnih polja. Izlazna vrijednost polja p nije bitna, tj. smijete ga preurediti u nešto različito od ulaza. Za bilo koji ulazni p, broj **zamjena** elemenata u polju A smije biti **najviše** jednak $n - 1$, a vremenska složenost cijelog algoritma mora biti u $O(n)$, tj. najviše linearna u n. Detaljno argumentirajte da vaš algoritam zadovoljava ove uvjete na složenost.

3. (20) Neka je $(A, \cdot)$ algebarska struktura polugrupe obzirom na binarnu operaciju množenja $\cdot : A \times A \to A$, tj. množenje je asocijativno i ima neutralni element $e \in A$. Za svaki element $a \in A$, korektno su definirane potencije a^n, za svaki $n \geq 0$. Složenost takvog potenciranja mjerimo brojem potrebnih množenja elemenata iz A. Množenje s neutralom e **ne brojimo**, jer se uvijek može izbjeći pažljivim testiranjem faktora!

(a) Napišite algoritam koji računa a^n, za zadane a i n, koristeći tzv. **binarno** potenciranje (pozicioni zapis broja n u bazi 2). Kolika je složenost tog algoritma, u ovisnosti o n?

(b) Pokažite primjerom da binarno potenciranje **ne mora** biti optimalno u pogledu broja množenja, tj. nađite neki n za koji se a^n može izračunati sa strogo manje množenja.

(c) Zadana je linearna homogena rekurzivna relacija reda k, s konstantnim koeficijentima

$$t_n = a_{k-1} t_{n-1} + \cdots + a_0 t_{n-k}, \quad n \geq k,$$

uz zadane početne uvjete $t_0, \ldots, t_{k-1}$. Ukratko opišite kako se binarno potenciranje može iskoristiti za **brzo** računanje n-tog člana t_n. Kolika je složenost u ovisnosti o n? Kolika je složenost u ovisnosti o k?

(d) Kako biste iskoristili binarno potenciranje za rekurziju

$$t_n = 2t_{n-1} - 3t_{n-2} + t_{n-3}, \quad n \geq 3,$$

uz $t_0 = 0, t_1 = 1, t_2 = 2$? Samo navedite bitne objekte, ne morate pisati cijeli algoritam.

REZULTATI: nedjelja, 4. 2. 2018., navečer (na webu),
ponedjeljak, 5. 2. 2018. u 12 sati. Saša Singer