OBLIKOVANJE I ANALIZA ALGORITAMA — 2. kolokvij

28. 1. 2015.

1. (20) Putnik treba prijeći dugački ravni put duljine L kilometara, s tim da svaki pojedini dan može prijeći najviše d kilometara. Pripremajući se za put, putnik je pronašao niz od n mogućih zaustavnih točaka gdje može prenoćiti, na strogo **rastućim** udaljenostima $x_1, \ldots, x_n$ od polazne točke. Za neki strogo rastući podniz zaustavnih točaka kažemo da je **korektan**, ako je udaljenost susjednih točaka najviše jednaka d, prva točka je udaljena najviše d od početka puta, a zadnja točka je udaljena najviše d od kraja puta. Drugim riječima, podniz je korektan ako putnik može stići do kraja puta, tako da svaki dan ne prelazi više od d kilometara i spava na zaustavnim točkama iz tog podniza. Pretpostavljamo da je polazni niz od svih n točaka korektan (ne treba provjeravati).

Treba naći **korektan** podniz polaznog niza zaustavnih točaka s **najmanjim** brojem elemenata, tako da putnik u najmanjem broju dana stigne na cilj.

(a) Dokažite da **pohlepni** izbor sljedeće zaustavne točke za svaki pojedini dan puta ("putuj najdalje što možeš") garantira korektnost i minimalnost dobivenog podniza.

(b) Sastavite algoritam koji nalazi traženi korektni podniz s najmanjim brojem elemenata. Algoritam treba vratiti **broj** elemenata i **polje indeksa** odgovarajućih zaustavnih točaka u tom podnizu. Složenost algoritma mora biti $O(n)$. Dokažite da vaš algoritam to zadovoljava.

OKRENITE!

2. (30) Zadan je skup od n objekata, numeriranih brojevima od 1 do n. Za svaki par objekata (i, j) poznata je njihova **udaljenost** $d(i, j)$, koja zadovoljava svojstva $d(i, i) = 0$ i $d(i, j) = d(j, i) > 0$, za $i \neq j$ (nejednakost trokuta nije bitna). Udaljenosti objekata zadane su simetričnom matricom D s elementima $d(i, j)$, za $i, j = 1, \ldots, n$. Ovih n objekata treba particionirati u k **nepraznih** disjunktnih grupa ili klastera $C_1, \ldots, C_k$, tako da svaki objekt pripada točno **jednoj** grupi. Bilo koju takvu particiju objekata u k nepraznih klastera zovemo k–klasteriranje. Za reprezentaciju particije koristimo strukturu **disjunktnih** skupova.

(a) Ukratko opišite što je struktura **disjunktnih** skupova, koje je polazno stanje strukture i koje su osnovne **operacije** definirane na toj strukturi (**što** rade te operacije, nije bitno **kako**).

(b) Opišite neku reprezentaciju strukture disjunktnih skupova, pripadne **efikasne** implementacije osnovnih operacija i komentirajte njihovu složenost.

Ideja grupiranja je da objekti unutar istog klastera budu "blizu", a da različiti klasteri budu što "dalji". Pojam **razmaka** nekog k–klasteriranja definiramo kao **najmanju** udaljenost bilo kojeg para objekata koji se nalaze u **različitim** klasterima. Za zadani broj k (uz $1 < k \leq n$) treba naći k–klasteriranje zadanih n objekata koje ima **najveći** mogući razmak.

(c) Sastavite algoritam koji nalazi takvo k–klasteriranje i vraća pripadne klastere $C_1, \ldots, C_k$ u izabranoj reprezentaciji disjunktnih podskupova skupa objekata $\{1, \ldots, n\}$.

Uputa: Pogledajte kako radi Kruskalov algoritam za nalaženje minimalnog razapinjućeg stabla **potpunog** grafa sa zadanim objektima kao čvorovima i zadanim udaljenostima.

(d) Kolika je složenost cijelog algoritma u funkciji od n? Koji dio algoritma ima dominantnu složenost? Ima li implementacija strukture disjunktnih skupova bitnog utjecaja na složenost?

3. (20) 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: ponedjeljak, 2. 2. 2015. u 14 sati. Saša Singer