AGOGE i-training
e-mail: agogeit@gmail.com
datum: 24. maj 2010. godine

Rešenja problema: Problem Set 03

Zadatak 01. Problem se radi **greedy** metodom. Kroz stablo ćemo se kretati počev od listova i na njihovim k-tim roditeljima postavljamo market i markiramo sve čvorove koji su udaljeni od njega za ne vişe od k čvorova. Postupak nastavljamo sve dok postoji nemarkirani čvor. Ovo vrstu obilaska stablo implementiramo preko **BFS**-a. Lako se može pokazati da je ovo ujedno i najbolje rešenje.

Zadatak 02. Ovo je jedan od težih problema iz teorije grafova. Ima dosta verzija ovog problema, medjutim svi se baziraju na istoj ideji. Dakle, rešenje tj. krajnji maksimalni količnik tražimo binarno. Kada ispitujemo konkretnu vrednost x, napravimo novi graf u kojem će težine ivica biti $a_e - x \cdot b_e$. Ispitivanje grane binarne pretrage (tj. da li x treba povećati ili smanjiti) svodi se na ispitivanje postojanja negativnog ciklusa u novom grafu - **Bellman ford algoritam**.

Zadatak 03. U ovom problemu se koristi nekoliko grafovskih algoritama. Na početku iskoristimo **Dijkstrin algoritam** za nalaženje najkraćih puteva od polaznog čvora. Zatim konstruišemo novi graf tako što uzmemo sve ivice koje su mogući kandidati za najkraće puteve i dodelimo im težinu 1. Nije teško videteti da rešenje postoji ako i samo ako u ovog grafu postoji **flow** veličine bar 2 (od kuće do škole). Za ovo koristimo neki od **MaxFlow algoritama** i ukoliko rešenje postoji, puteve lako rekonstruišemo pomoću dva BFS-a. Ukupna složenost je $O(n^2)$ jer nas interesuje samo da li je protok veći ili jednak 2.

Zadatak 04. Podelimo graf na **jako povezane komponente**. Novi graf u kome su ove komponente čvorovi je acikličan. Zbog toga, ukoliko je čvor u *sink*, komponenta u kojoj se on nalazi ne sme imati izlaznih grana jer je onda nemoguće vratiti se u u. Sa druge strane, ukoliko neka komponenta nema izlaznih grana, svi čvorovi u njoj su *sink*-ovi, pa su rešenje svi čvorovi iz ovih komponenti. Složenost je $O(n+m)$.

Zadatak 05. Ključna obzervacija je da nam se uvek najviše isplati da markiramo grane koje su bliže korenu, jer onda povećavamo broj markiranih grana i u odgovarajućem podstablu. Prema tome, **DFS** obilaskom krenemo od korena i za svaki čvor pamtimo broj makiranih grana na putu do njega i samu dužinu puta, na osnovu čega vidimo da li trenutnu granu treba markirati ili ne. Složenost je $O(n)$.

Zadatak 06. Zadatak rešavamo **dinamičkim programiranjem nad stablom**. Za svaki čvor pamtimo najduži put od njega do nekog lista u njegovom podstablu i najduži put preko njegovog direktnog pretka. Za prvi podatak stablo treba obići odzdo nagore a za drugi odozgo nadole. Pri tome treba čuvati i drugi najduži put u podstablu, ukoliko se ispostavi da iz direktnog pretka naduži put vodi preko posmatranog čvora. Složenost je $O(n)$.

Zadatak 07. Glavna obzervacija pri rešavanju problema jeste da nama treba samo jedan poligon za optimalno rešenje. Algoritam možemo podeliti u dva dela: konveksni omotač i traženje najmanjeg puta. Prvo nalazimo **konveksni omotač** rupa i za svako drvo ispitujemo da li je unutra ili van.

Naravno ukoliko nijedno drvo nije unutra, kao rešenje vraćamo $111M$.

Sada kreiramo usmereni graf u kojem su čvorvi rupe, a ivice parovi rupa. Ivica (v_1, v_2) postoji akko je svo drveće u omotaču sa leve strane ivice. U ovako kreiranom grafu treba naći ciklus sa minimalnim brojem ivica. Ovo se može uraditi uz pomoć **Floyd-Warshall algoritma**.

Složeno kreiranja konveksnog omotača je $O(n \log n)$ uz pomoć **Graham scan-a**, dok provera da li je tačka unutra ili ne zahteva $O(n)$. Inicijalizacija grafa i nalaženje ciklusa imaju složenot $O(n^2 \cdot m)$ tj. $O(n^3)$, što nas dovodi do složenosti $O(n^3 + n^2 \cdot m)$.

Zadatak 08. Ovo je jako simpatičan problem. Ukoliko solidno poznajete algoritme iz teorije grafova, možete odmah zaključiti da se ovaj algoritam može svesti na analizu **artikulacionih čvorova** i **mostova**. Nalaženje istih u grafu rešavamo uz pomoć algoritma za podelu grafa na **bipovezane komponente (eng. biconnected components)**. Nakon kreiranja **BCC stabla** iz upiti se svode na ispitivanje jedinstevnog puta izmedju čvorova, tj. da li tom putu pripada uklonjena ivica ili čvor (naravno samo u slučaju da je ona most odnosnoo artikulacioni čvor).