Tema 3

Algoritmi za rješavanje svih parova najkraćih puteva u usmjerenom težinskom grafu.

Svi parovi najkraćih puteva osnovni algoritam

Na osnovu grafa predstavljenog s matricom W, računamo niz matrica L(NxN) u funkciji

Extend-shortest-paths , koja računa najkraći put čvorova dosad. Gleda da li je predhodni pronađen put manji od puta koji ide preko nekog čvora k.

Vrijeme izvršavanja algoritma je O(N^4) jer funkcija extend-shortest-paths radi sa tri ugniježdene for petlje, a pozivamo ja n puta.

Floyd Warshall

Floyd-Warshall Algoritam je algoritam za pronalaženje najkraćeg puta između svih parova čvorova u posmatranom grafu. Ovaj algoritam radi i za usmjerene i neusmjerene grafove ali ne radi za grafove sa negativnim ciklusima. Kao rezultat ovog algoritma dobijamo matricu (NXN) koja predstavlja najkraći putevi od I-tog do j-tog čvora.

Ideja je da popunjavamo matricu tako što uzima se minimum između predhodno izračunatog puta za dati put ili zbir najkraćih puteva od početnog čvora preko nekog čvora k I puta od čvora k do krajnjeg čvora.

Kompleksnost algoritma je O(V^3), zbog tri ugniježdene for petlji a sve unutar njih obavlja se u konstantnom vremenu.

25.1-10 Broj čvorova u najmanjem negativnom ciklusu grafa

Algoritam radi u vremenu O(N^4) jer je zasnovan na prvom algoritmu za pronalazak svim parova najkraćih puteva,uz dodatnu provjeravu O(N^2) da li na glavnoj dijagonali ima negativnih brojeva čim se desi da postoji vraćam broj koliko se ponavljala petlja toliko je čvorova u negativnom ciklusu.

Johnsonov algoritam

Ideja algoritma je da kreiramo novi graf G' , koji ima jedan čvor više od grafa G. Taj graf ima iste grane kao I graf G, I još n dodatnih grana sa početkom u novom vrhu ,a krajem u ostalih n vrhova čija je težina grana 0. Zatim pokrene se Bellman Fordov algoritam nad G' sa početnimi čvorom s. Ukoliko on prepozna da postoji negativan ciklus u G', to će značiti da negativan ciklus postoji I u grafu G. Potrebno je onda da napravimo graf takav da nema negativnih grana, ali da mu najkraći putevi ostanu iste vrijednosti da bismo mogli pozvati Dijkstra algoritam nad tim grafom. To se postiže tako da svakom čvoru dodijelimo neku vrijednost p(v), v iz G ta vrijednost je ustvari vrijednost najkraćeg put od s do tog čvora. Nakon toga svakoj grani promjenimo vrijednost na nova_tezina=stara_tezina+p(pocetak)-p(kraj) na taj način dobijem graf sa nenegativnim grama. Pozovem Dijkstra algoritam nad svakim čvorom grafa G I dobijem najkraće puteve koje kasnije moram postaviti na najkraći put orginalnog grafa G. Imali smo da je nova_tezina=stara_tezina+p(pocetak)-p(kraj), sada nakon Dijsktra algoritma u vektoru (v) koji on vrati imamo najkraće puteve za graf s novim težinama . Najkraći putevi orginalnog grafa rekonstruisem , tako sto samo oduzmem p(pocetak) I saberem p(kraj) i ove vrijednosti spremam u matricu NxN, koju vraćam iz funkcije.

Kompleksnost

Osnovni koraci su poziv Belman Fordovog algoritma jednom I Dijsktra algoritma za svaki čvor.

Belman Ford ima kompleksnost O(EV) a Dijkstra ima kompleksnost O(VlogV)

Pa kompleksnost Johnsonovog algoritma O(V^2 logV + EV).

25.2-8 Algoritam koji računa u vremenu O(EV) da li postoji put između svaka dva čvora usmjerenog grafa.

Algoritam se zasniva na DFS pretrazi, na način da poziva DFS nad svakim čvorom grafa G I označava ih da su posjećeni zatim se DFS poziva nas tim čvorom I susjedima od drugog proslijeđenog čvora ukoliko nije posjećen.

Obzirom da DFS radi u vremenu O(E+V) I poziva se onoliko puta koliko ima čvoroa tj. V puta, onda je kompleksnost O(V(E+V)).