Konstrukcija i analiza algoritama
6. čas

Strahinja Stanojević

26.03.2020.

1. Koristeći BFS pretragu grafa pronaći najkraći put izmedju čvora u i čvora v u neusmerenom grafu G.

2. Napisati iterativnu funkciju za ispis binarnog stabla pretrage po nivoima.

3. Implementirati Kanov algoritam za topološko sortiranje grafa kao i algoritam zasnovan na DFS-u.

4. Modifikovati algoritam za topološko sortiranje grafa tako da ukoliko nema ciklusa ispisuje topološko sortitranje, inače ispisuje neki ciklus.

5. Student da bi mogao izaći na ispite koje je želeo da polaže mora da obezbedi saglasnost profesora da je pratio njihova predavanja. Ali, neki profesori daju saglasnost samo ako student je pribavio saglasnost od profesora čija predavanja su preduslov za razumevanje njegovog predmeta. U prvoj liniji standardnog ulaza dat je broj profesora N (0<N<101) od kojih student trazi potpise. U narednih N redova date su potrebne informacije tako da prvi broj K u liniji I+1 predstavlja broj profesora čiji potpisi su preduslov za potpis profesora I. Narednih K brojeva označavaju te profesore. Ako se ne mogu prikupiti potpisi svih profesora, onda na standardni izlaz treba ispisati poruku NE, a ako se mogu prikupiti svi potpisi onda ispisati poruku DA u 1. red standardnog izlaza, a u narednih N linija ispisati redosled prikupljanja potpisa.

6. Neka je dat aciklički graf G koji ima i usmerene i neusmerene grane. Konstruisati algoritam koji usmerava neusmerene grane tako da graf ostane aciklički.

7. Neka je dat aciklički usmeren graf G. Konstruisati algoritam koji dodaje maksimalan broj grana u ovom grafu tako da on i dalje ostane acikličan.

8. Konstruisati algoritam za odredjivanje svih artikulacionih tačaka u neusmerenom grafu G.

9. Konstruisati algoritam za odredjivanje svih mostova u neusmerenom grafu G.

10. Konstruisati algoritam koji pronalazi čvor koji se može izbaciti iz neusmerenog grafa G tako da graf ostane povezan.

11. Konstruisati algoritam koji u neusmerenom grafu G pronalazi sve cikluse dužine n.

12. Za neusmereni graf G kažemo da je bipovezan ako važi da za bilo koja 2 čvora u i v postoje 2 različite putanje kojima su oni povezani, tj ako iz u do v možemo doći na 2 načina. Konstruisati algoritam koji za dati graf proverava da li je bipovezan. Može se pretpostaviti da ne postoje više od dva puta izmedju bilo koja dva čvora. (Bolji tekst bi mozda bio bar 2 načina).