

**CS103 - DOMAĆI ZADATAK 11.**

Prilikom slanja domaćeg zadatka svom asistentu neophodno je da ispunite sledeće:

* Subject mail-a mora biti CS103-DZbr. Za ovaj domaći zadatak - CS103-DZ11
* Sve fajlove, koji su deo rešenja zadataka, arhivirati (zip, rar, …)
* Poželjno je uraditi i printscreen koda pre pokretanja programa
* U prilogu maila treba da se nalazi arhiva projekta koji se ocenjuje, imenovana na sledeći način: CS103-DZbr-BrojIndeksa-ImePrezime*. Na primer, CS103-DZ11-9999-VeljkoGrkovic.zip*
* Telo mail-a treba da ima pozdravnu poruku

**Molimo sve studente da se pridržavaju navedenog, inače zadaci neće biti pregledani i ocenjeni.**

Studenti iz Beograda i online studenti zadatke na ocenjivanje šalju mail-om na adresu [lazar.mrkela@metropolitan.ac.rs](mailto:lazar.mrkela@metropolitan.ac.rs)

Studenti iz Niša zadatke na ocenjivanje šalju mail-om na adresu  
[nikola.dimitrijevic@metropolitan.ac.rs](mailto:nikola.dimitrijevic@metropolitan.ac.rs)

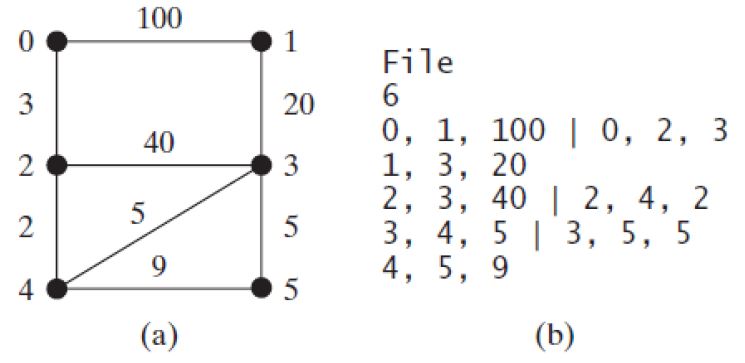
**Rok za slanje domaćih zadataka:**

* **Tradicionalni studenti: sedam dana od dana održavanja vežbe na koju se odnosi domaći zadatak. Nakon toga, broj poena se umanjuje za 50%. Krajnji rok za slanje rešenja je deset dana pre termina ispita u kome želite da polažete ispit.**
* **Studenti na online nastavi, kao i studenti kojima je predmet prenet u narednu godinu ili je diferencijalni: najkasnije deset dana pre termina ispita u kome želite da polažete ispit.**

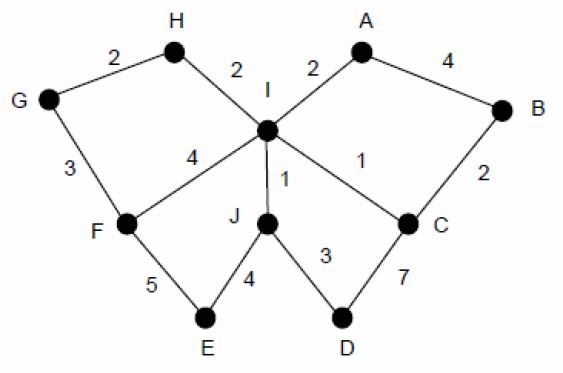
*Svaki student radi jedan zadatka sa spiska*. Ostali zadaci mogu da posluže za vežbanje i pripremu ispita, ali ih ne šaljete na pregled asistentima.

Redni broj zadatak dobijate tako što vaš broj indeksa podelite sa 18, a dobijeni ostatak pri deljenju uvećate za 1. Na primer:   
Broj indeksa 2378  
2378 % 18 + 1 = 3 – student radi 3. zadatak.

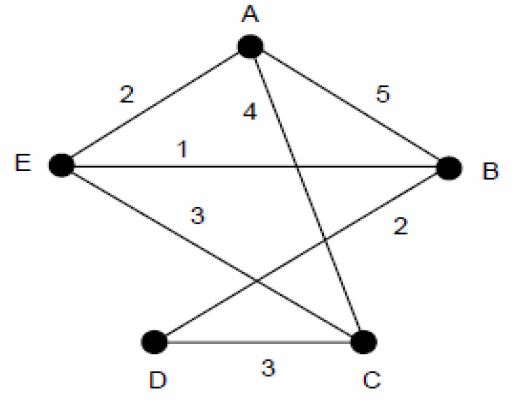
1. Napisati program u Java-i za implementaciju Kruskalovog algoritma korišćenjem reda prioriteta. Red prioriteta služi za određivanje prve naredne grane koja ima najmanju težinu i koja će biti razmatrana u cilju ubacivanja u minimalno povezujuće stablo.
2. Napisati program za implementaciju Dijkstra algoritma korišćenjem reda prioriteta za susedne grane, kao i korišćenjem matrice susedstva koja pamti težine odgovarajućih grana.
3. Napisati program koji učitava povezani graf iz fajla i štampa njegovo minimalno povezano stablo. Prva linija fajla se sastoji iz broja temena (n). Temena su označena brojevima 0, 1, ..., n-1. Svaka naredna linija sadrži podatke o granama koje izlaze iz odgovarajućeg čvora u obliku u1, v1, w1 | u2, v2, w2 | .... Svaki triplet predstavlja granu (određenu sa u i v) i težinu grane w. Pretpostavka je da je graf neusmeren, pa stoga ako ima granu (u, v) onda će imati i granu (v, u). Samo jedna grana je predstavljena u fajlu, pa se stoga i druga grana treba uzeti u obzir prilikom određivanja minimalnog povezujućeg stabla.



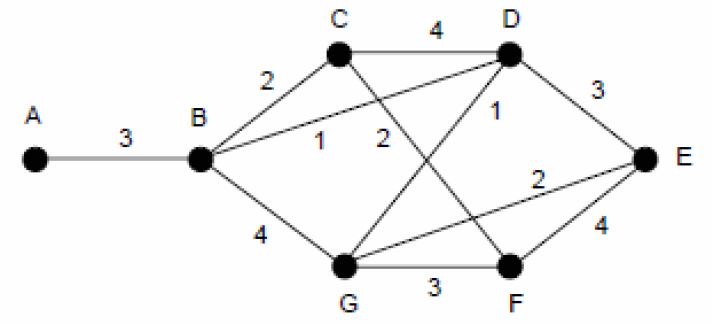
1. Student treba da sluša određen broj kurseva da bi diplomirao, i svaki od ovih kurseva ima preduslovne kurseva koji moraju biti slušani pre izbora drugog kursa. Pretpostavka je da su svi kursevi dostupni bilo kog semestra i da student može da sluša neograničen broj kurseva. Ako je data lista kurseva i spisak svih njegovih preduslovnih kurseva (kurs V ima preduslovne kurseve U1,..,Un), odrediti raspored koji zahteva minimalan broj semestara kako bi student mogao da diplomira. Rešiti problem korišćenjem topološkog sortiranja. Kao ulaz je dat usmereni netežinski graf.
2. Pretpostavimo da imate graf u kome grane predstavljaju direktnu vezu između dva računara. Svaka grana (v,w) ima tezinu pv,w koja predstavlja verovatnoću da će prenos između računara v i w uspeti (0 < pv,w < 1). Napisati program koji će za izabrani računar S pronaći najpogodniju putanju za prenos podataka do svih ostalih računara u mreži.
3. Modifikovati Primov algoritam da umesto minimalnog pronalazi maksimalno sprežno stablo. Napisati program u kome se vrši unos grafa iz fajla (broj temena, broj grana, i za svaku granu uređena trojka – cvor1, cvor2, težina grane).
4. Najkraća putanja u mreži. Ako je data matrica NxN sa pozitivnim celim brojevima, pronaći najkraći put od polja (0, 0) do polja (N-1, N-1) pri čemu je ukupan put zbir brojeva na putanji (može se kretati samo levo, desno, dole, gore - ne može dijagonalno).
5. Najkraća bitonična putanja. Ako je dat usmeren težinski graf, pronaći najkraću bitoničnu putanju od početnog čvora S do ostalih čvorova u grafu. Putanja je bitonična ako postoji prelazni čvor (međučvor u putanji) V takav da su sve grane na putu od S do V rastuće, a sve grane na putu od V do T strogo opadajuće. Putanja mora biti jednostavna (nema ponavljanja čvorova u putanji).
6. Najkraća putanja u mreži. Ako je data matrica NxN sa pozitivnim celim brojevima, pronaći najkraći put od polja (0, 0) do polja (N-1, N-1) pri čemu je ukupan put zbir brojeva na putanji (može se kretati samo desno i dole.
7. Za graf sa slike pronaći minimalno razapinjajuće stablo Kruskalovim algoritmom. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



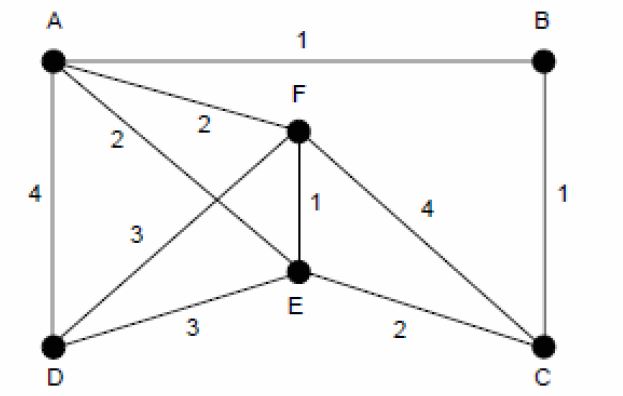
1. Za graf sa slike pronaći minimalno razapinjajuće stablo Primovim algoritmom. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



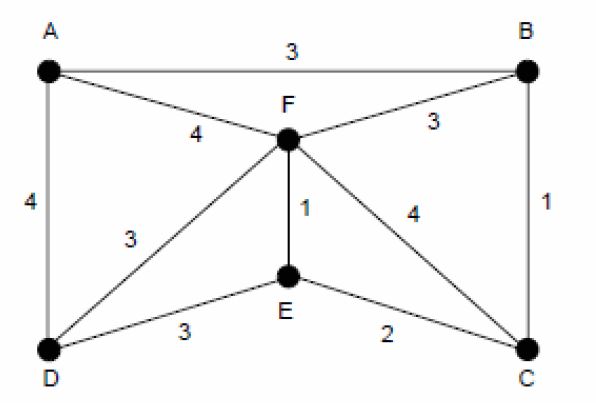
1. Za graf sa slike pronaći minimalno razapinjajuće stablo Primovim algoritmom. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



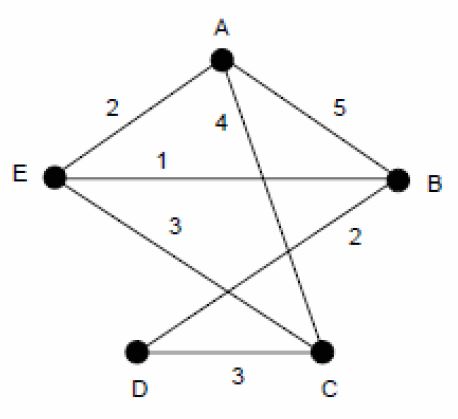
1. Za graf sa slike pronaći minimalno razapinjajuće stablo Kruskalovim algoritmom. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



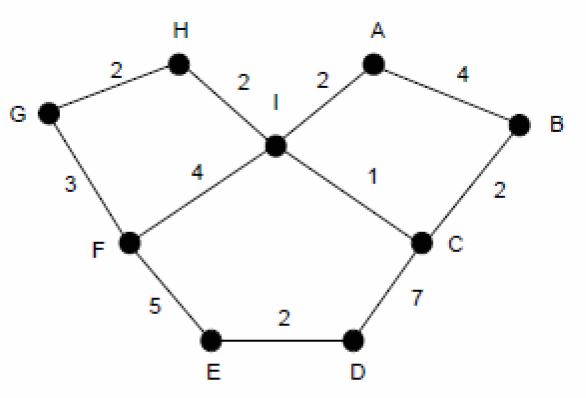
1. Za graf sa slike pronaći najkraće puteve od čvora A do svih ostalih čvorova u grafu. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



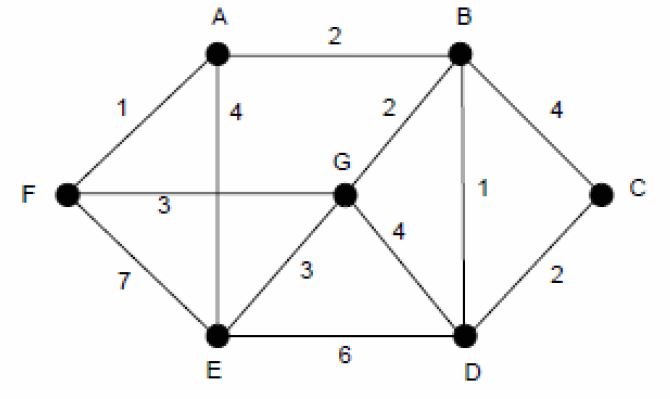
1. Za graf sa slike pronaći najkraće udaljenosti između svih čvorova u grafu. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



1. Za graf sa slike pronaći najkraće puteve od čvora D do svih ostalih čvorova. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



1. Za graf sa slike pronaći najkraće puteve od čvora C do svih ostalih čvorova. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.



1. Za graf sa slike pronaći najkraće puteve od čvora E do svih ostalih čvorova. Graf se učitava iz fajla u kome prva linija sadrži broj čvorova i broj grana u grafu, a svaka naredna linija sadrži uređene trojke (u,v,w), gde u i v predstavlja granu između ta dva čvora, a w težine. Program treba da štampa svaki prolaz, do konačnog rešenja.

