

# Napredni algoritmi i strukture podataka

## 3. laboratorijska vježba

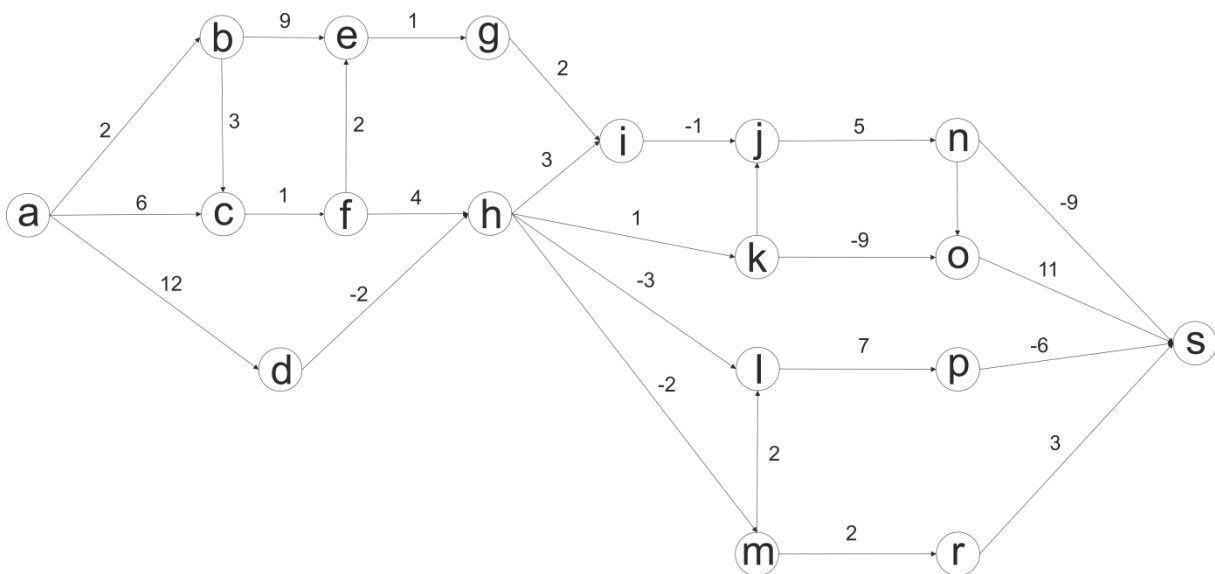
Napomena: umjesto navedenih zadataka možete raditi i nešto drugo vezano uz gradivo trećeg niza predavanja, ali to mora biti odgovarajuće složenosti i u dogovoru s Vašim asistentom.

### Zadatci za 5 bodova

- 1) Modelirati neki problem linearnim programom i riješiti ga Simplex algoritmom.
  - simplex algoritam nije potrebno programirati, slobodno koristite Matlab programe dostupne na stranici predmeta. Naravno, morat ćete objasniti pripremu podataka za daljnju obradu.
  - za kolokviranje vježbe važno je pregledno i jasno opisati i postaviti linearni optimizacijski problem
  - prikazati (može copy-paste iz programa) dva-tri koraka algoritma te analizirati i komentirati promjene rješenja

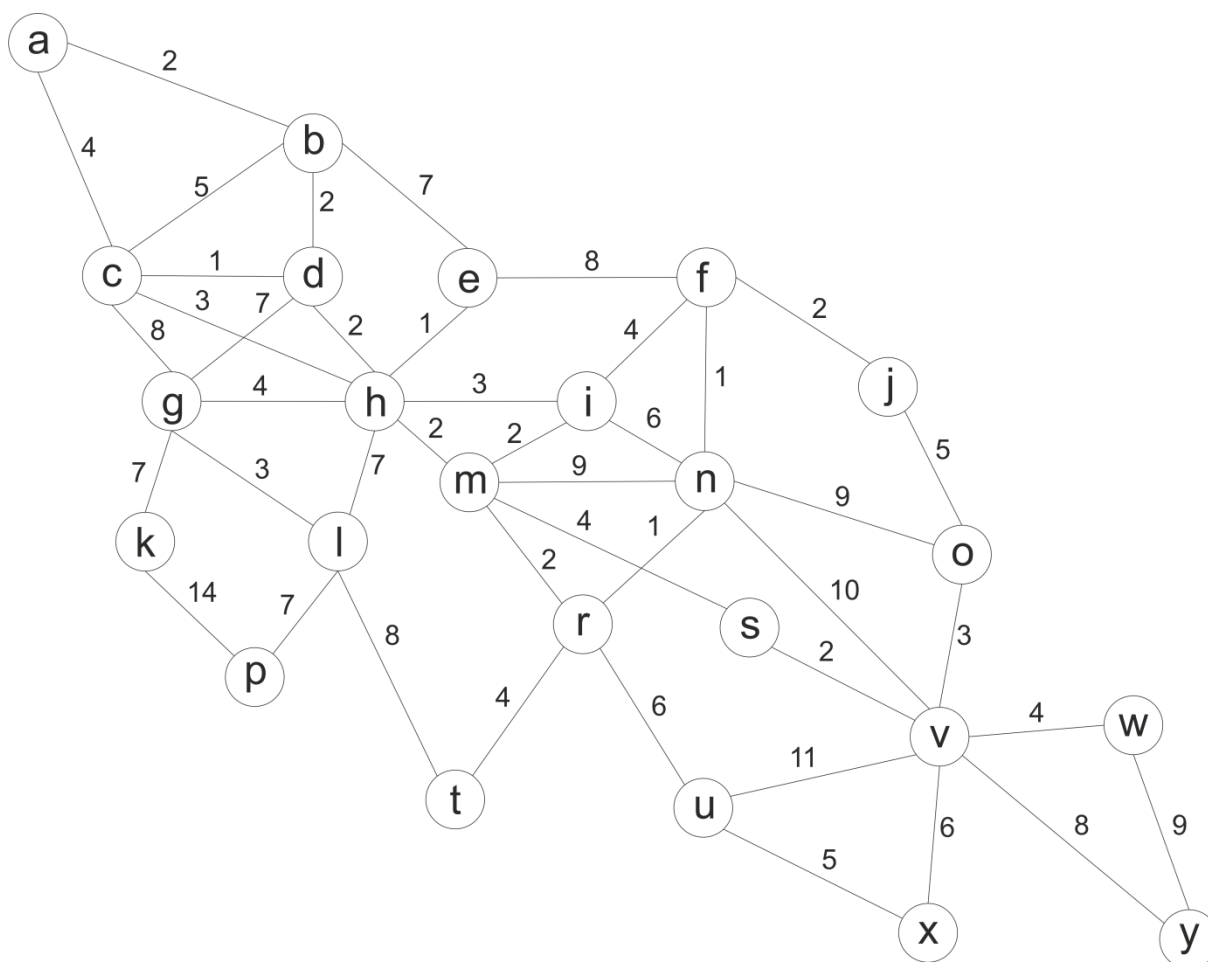
### Zadatci za 11 bodova

- 1) Riješiti zadatak sa simpleks algoritmom iz skupine za 5 bodova, ali pisanjem vlastitog programa (C#, C++, Java).
- 2) Modelirati grafom dio nekog naselja i programski odrediti najkraći put između dva mjesta (dva vrha). Tko želi, može modelirati i nešto drugo, gdje bi bilo čak i bridova negativnih težina.
  - za kolokviranje vježbe važno je pregledno i jasno opisati model te organizaciju podataka u programu
  - nije potrebno graditi komplicirane modele, dovoljni su grafovi s 10...20 vrhova. Naravno, kompliciraniji modeli će vjerojatno biti i izazovniji te kao takvi zanimljivija i „plodonosnija“ laboratorijska vježba.
  - program mora riješiti pohranu grafa u kompjutoru, pronalaženje najkraćeg puta i ispis (iscrtavanje) rješenja
  - iscrtavanje grafa i najkraćeg puta nije obavezno, nego samo poželjno, ali prikladan ispis najkraćeg puta je obavezan
  - iscrtavanje se brzo i relativno lako može postići prepuštanjem tog posla slobodnom (*open source*) programu *Graphviz* koji možete preuzeti sa stranice <http://www.graphviz.org/>, gdje su i detaljne upute za njegovo korištenje. Dovoljno je iskoristiti samo njegovu osnovnu funkcionalnost, bez posebnog dotjerivanja rješenja, a ni njegovo pozivanje ne mora biti automatsko, nego je dovoljno programski pripremiti podatke za *Graphviz*, a pozivati ga možete i „ručno“ iz komandne linije.
  - tko ne zna što modelirati, može raditi s grafom na slici



- 3) Pomognite svojem poštaru! Modelirajte grafom dio naselja koji Vaš poštar mora obići i predložite mu najkraći obilazak svih ulica za koje je zadužen (programski riješiti Chinese Postman Problem).
- vrijede iste napomene kao i za prvu predloženu vježbu, s time da treba prikladno ispisati najkraći obilazak
  - možete se suočiti s problemom uparivanja (*matching*) čije rješavanje nije trivijalno i iziskuje dosta programiranja. Konkretno problem će vjerojatno biti moguće riješiti „ručno“ pa je dovoljno pripremiti skice (može iscrtane rukom, ne moraju biti u elektroničkom obliku) i tijekom kolokviranja objasniti postupak. Bude li problem prekomplikiran za ručno rješavanje, a nemate vremena i/ili volje proučavati i programirati odgovarajuće algoritme, predlažemo nekoliko mogućih izlaza:
    - (naša preporuka) preinačite model tako da se problem dovoljno pojednostavi
    - iskoristite svoju prirodnu inteligenciju i intuiciju te sami pronađite najbolje parove. Problem je samo kako ćete sebi i nama dokazati da je predloženo rješenje stvarno najbolje!
    - isprogramirajte ispitivanje svih mogućih kombinacija pa imate li dovoljno brz kompjutor i ne prekomplikirani graf, do kraja semestra možda i dobijete rješenje ☺
    - nemojte raditi ovu vježbu. ☺

- 4) Za graf na slici programski pronaći najmanje razapinjajuće stablo (*minimum spanning tree* - MST) pomoću sva tri algoritma obrađena na predavanjima (Kruskalov, Dijkstrin i Primov algoritam).
- detekciju ciklusa riješiti po volji, a odabir i izvedba algoritma će svakako biti predmet razgovora tijekom kolokviranja
  - rezultat (MST) treba prikladno ispisati ili vizualizirati



**Zadatci za 17 bodova :** posebno razrađena, domišljata i uspješna rješenja zadataka iz skupine za 11 bodova. Procjena asistenta!