

FINALNI ISPIT

13 Jun 2019

NAPOMENE

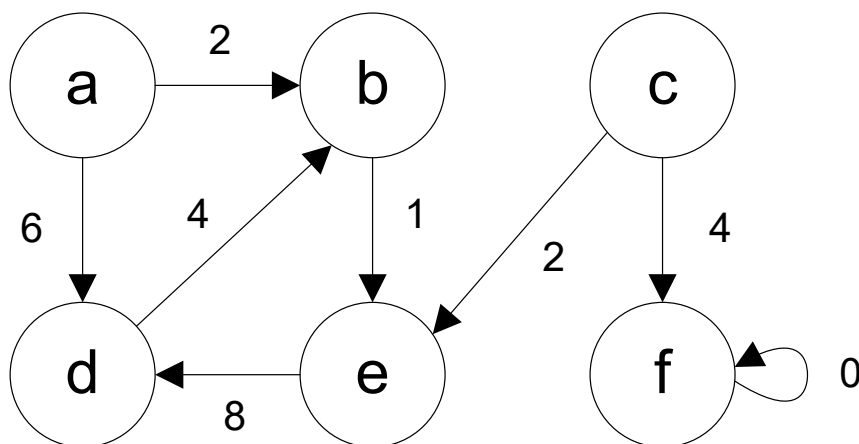
Za potrebe testa napraviti direktorijum **C:\tmp\student\X**, gde je X broj indeksa. Rešenje zadatka treba da se nalazi u tom direktorijumu.

Direktorijum **student\X** zajedno sa rešenjem treba arhivirati i okačiti na lični intranet portal. Direktorijum **C:\tmp\student\X** ne treba brisati.

ZADATAK

Za prikazani usmereni graf sa slike odrediti jako povezane komponente (engl. Strongly Connected Components). Jako povezane komponente grafa se određuju algoritmom koji se zasniva na pretrazi u dubinu (engl. Depth-First Search). Koraci algoritma su sledeći:

- Pretražiti ulazni graf i odrediti vremena završetka pretrage svakog čvora
- Transponovati graf
- U opadajućem redosledu u odnosu na vremena završetka pretrage svakog čvora ponoviti pretragu ali na transponovanom grafu
- Ispisati sve elemente stabala koji čine šumu stabala formiranu nakon poslednje DFS pretrage



Slika 1. Primer usmerenog grafa

1. Osnovni koraci algoritma treba da budu implementirani u funkciji *ApplyStronglyConnectedComponents*.
def ApplyStronglyConnectedComponents() --> returns None
2. Napisati funkciju *MakeGraph* koja formira graf sa slike i vraća ga kao povratnu vrednost.
def MakeGraph() --> returns graph
3. Napisati funkciju *TransposeGraph* koja transponuje ulazni graf. Transponovani graf je graf koji ima obrnute smerove svih veza u odnosu na ulazni graf. Povratna vrednost funkcije je transponovani graf.
def TransposeGraph(graph) --> returns graph
4. Napisati funkciju *ApplyDFS* koja primenjuje pretragu u dubinu.
def ApplyDFS(graph) --> returns None
5. Napisati funkciju *ApplyDFSInOrder* koja primenjuje pretragu u dubinu po opadajućim vremenima završetka pretrage čvorova.
def ApplyDFSInOrder(graph) --> returns None
6. Napisati funkciju *PrintStronglyConnectedComponents* koja ispisuje jako povezane komponente grafa, tj. elemente stabla koji čine šumu stabala.
def PrintStronglyConnectedComponents(graph) --> returns None