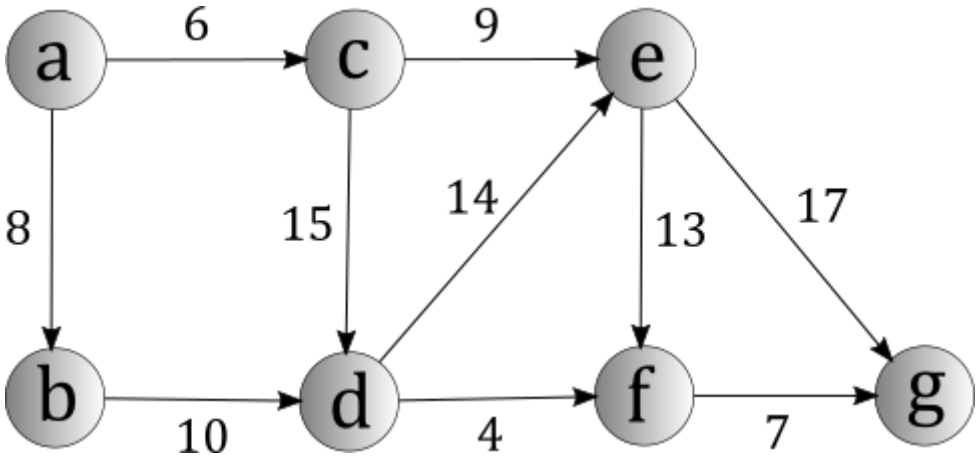
ZADATAK

1. Napisati funkciju *MakeGraph* koja formira graf sa slike 1.



Slika 1 – Primer grafa.

2. Preklopiti operator *__str__* klase grafa tako da ispisuje graf u obliku:
   `src ivica -> dst ivica, w = težina`

3. Napisati funkciju *GetInDegrees* i *GetOutDegrees* koje računaju ulazne i izlazne stepene (rangove) svakog čvora u grafu. Povratna vrednost obe funkcije je uređeni par. Prva vrednost je lista stepena čvorova koji su raspoređeni u istom redosledu u kome se nalaze u ulaznom grafu. Druga vrednost je rečnik gde je ključ ime čvora, a vrednost lista stepena tog čvora.
   `def GetInDegrees(graph)  --> returns (List, Dict)`
   `def GetOutDegrees(graph)  --> returns (List, Dict)`

4. Napisati funkciju *ShortestPath* koja računa najkraću putanju u datom grafu između dva čvora koja se prosleđuju kao parametar. Povratna vrednost je **par vrednosti (najkraća putanja, dužina)**. Koristeći ovu funkciju izračunati najkraću putanju između čvorova A i G sa grafa sa slike iz zadatka (1).
   def ShortestPath(graph, nodeA, nodeB) --> returns (List, int)

5. Napisati funkciju *UpdateEdge* koja dodaje ivicu između dva čvora sa težinom koja se prosleđuje kao ulazni parametar. Ukoliko ivica već postoji između ta dva čvora, ova funkcija menja težinu date ivice.
   `def UpdateEdge(graph, nodeA, nodeB, weight)`

6. Modifikovati graf napravljen u zadatku (1) koristeći funkciju iz zadatka (4) **dodajući ivicu od čvora B do čvora C sa težinom $w(B, C) = -4$**.

7. Modifikovati graf napravljen u zadatku (1) koristeći funkciju iz zadatka (4) **dodajući ivicu od čvora D do čvora E sa težinom $w(D, E) = -10$**. Pomoću funkcije *ShortestPath* implementirane u zadatku 4, izračunati najkraću putanju između čvorova A i G.