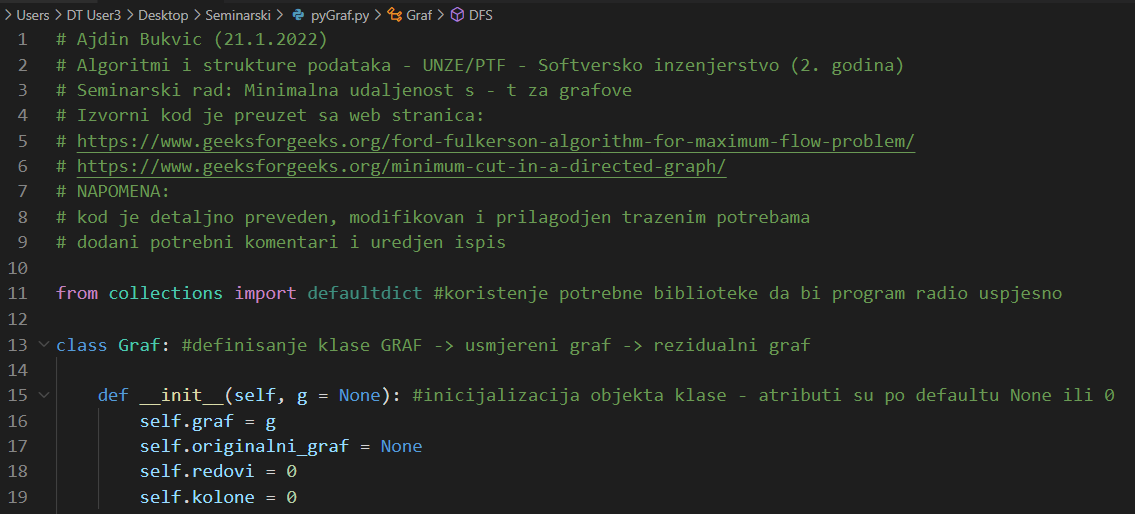
**Implementacija u Pythonu**

Za realizaciju algoritma koji proračunava minimalni s – t odsječak u traženoj protočnoj mreži, prvo je bilo potrebno kreirati klasu graf, s osnovnim metodama koje su potrebne za uspješno izvršavanje algoritma. Konstruktor klase graf se sastoji od atributa: graf, originalni graf, redovi, kolone. Graf je zapravo predstavljane kao težinska matrica formata n x n (kvadratna matrica) koja u sebi sadrži podatke o težinama između svih pojedinačnih vrhova. Originalni graf je zapravo kopija težinske matrice grafa, koja će biti potrebna prilikom daljnih provjera. Bitno je da se sačuvaju originalne težinske vrijednosti, zbog toga što će se primjenom algoritma, te vrijednosti mijenjati u zavisnosti od slučaja. Redovi i kolone daju samo informaciju o koliko se redova i kolona radi u matrici, pošto je u pitanju kvadratna matrica, te pošto smo se u ovom primjeru odlučili na jedan unaprijed određen primjer, ovdje se radi s grafom koji ima 6 redova i 6 kolona.



Metode koje posjeduje klasa graf su:

* *BFS (Breadth first search)*

Algoritam BFS – traženje po širini se sastoji od pretpostavke da su svi vrhovi na početku neposjećeni. Prvo se kreira pomoćni red u koji se sprema prvi element i označava se kao posjećen (izvor). Zatim se prolazi kroz while petlju dok god ima elemenata u redu. Zatim se uzima element iz reda, te se posjećuju svi njegovi susjedni vrhovi. Ova operacija se izvršava, tako što se onda dalje traže i svi „susjedovi“ susjedni vrhovi. Kada se svi neposjećeni vrhovi posjete, onda se provjerava da li postoji put od izvora do ponora, te se u slučaju pronalaska vraća True, ili u slučaju nepostojanja puta False.

* *DFS (Depth first search)*

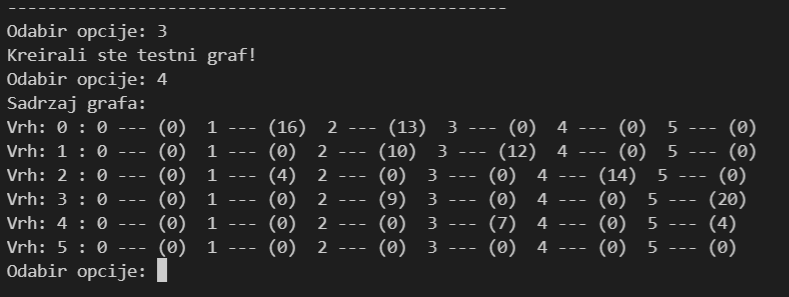
Algoritam DFS – traženje po dubini, prvo kreće postavljanjem posjećenosti izvora na True. Nakon toga se rekurzivnim pozivima metode DFS posjećuju njegovi susjedi. Kada se prođe kroz sve susjedne vrhove, odnosno kada svi postanu označeni, to signalizira prekid rekurzivnih poziva i završetak funkcije.

* *kreirajGraf*

Metoda kreirajGraf na jednostavan način kreira težinsku matricu nekog grafa. U našem konkretnom primjeru nije potrebno ništa unositi, jer je matrica statički definisana u samom kodu, uspješno kreiranje matrice i postavljanje atributa objekta klase graf, rezultira porukom da je kreiranje matrice bilo uspješno.

* *ispis*

Metoda za ispis prolazi kroz težinsku matricu datog grafa, te ispisuje njegove vrhove (0-5), te težinske vrijednosti između pojedinačnih vrhova. Iz ispisa se lahko može primjetiti i zaključiti koju težinu ima neka ivica, te koja 2 vrha su incidentni sa tom ivicom, odnosno koja ivica spaja 2 vrha.

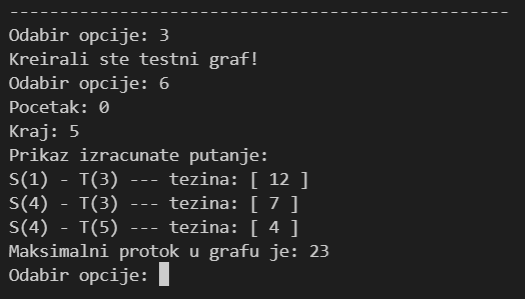


* *FordFulkerson*

Algoritam Ford-Fulkerson je usko povezan s algoritmom pronalaska minimalnog s – t odsječka za grafove. Ovaj algoritam se koristi za rješavanje problema maksimalnog protoka. U našoj konkretnoj realizaciji imamo sljedeće karakteristike: graf predstavlja protočnu mrežu, svaka ivica ima svoj kapacitet (težinu), traži se maksimalni protok između dva vrha s – t (izvor i ponor). Kreće se od pretpostavke da je početni protok jednak nuli. Korištenjem while petlje i uslova BFS, sve dok postoji put od izvora do ponora proračunava se prošireni put od izvora do ponora, te se on dodaje na maksimalni protok. Iteracijama kroz petlju se dodaju putanje i povećava protok, tako da se maksimalni protok postiže onda kada više ne postoji putanja koja je u stanju povećavati protok u grafu. Na kraju računanja funkcija treba da vrati i ispiše maksimalni protok.

* *MinCut*

Pronalazak minimalnog s – t odsječka se zasniva na korištenju Ford-Fulkerson algoritma. Ova dva algoritma su povezani teoremom koji kaže da je količina maksimalnog protoka u protočnoj mreži jednaka kapacitetu minimalnog odsječka. Uz primjenu Ford-Fulkerson algoritma dobija se kapacitet minimalnog odsječka, upravo iz tog razloga u ovoj metodi prvo je potrebno pozvati metodu FordFulkerson. Poslije toga se poziva DFS algoritam, gdje se na početku pretpostavi da su svi vrhovi neposjećeni. Onda se pronalazi skup vrhova koji su dostupni iz izvora, a sve one ivice koje su od dostupnog do nedostupnog vrha predstavljaju ivice minimalnog odsječka. Na kraju su sve te ivice ispisane, zajedno sa svojim vrijednostima kapaciteta (težine).



* *main.py*

U glavnom programu se prvo importuje prethodno kreirani fajl s implementacijom grafa i algoritama (pyGraf.py). Potrebno je prije svega kreirati jednu instancu klase graf nad kojom će se moći pozivati implementirane metode. Glavni program se sastoji od jednog menija, koji ima opcije za poziv metode koje posjeduje klasa graf. Funkcija meni se sastoji od standardne match-case strukture. Ukupno ima 6 opcija, dok je za prekid programa izlazna opcija broj nula. Meni je realiziran na klasičan način unosom opcije (gdje se forsira tip podatka „int“), zatim kroz while petlju se izvršava sve dok korisnik ne odluči prekinuti program (unosom broja 0).

