**UPRAVLJANJE ZNANJEM**

Završni ispit – praktični dio

Kolegij: Upravljanje znanjem

Nositelj kolegija: *izv. prof. dr. sc. Ana Meštrović*

Asistent:  *dr. sc. Slobodan Beliga*

Izradio: Vedran Orešković

Datum izrade: 12.2.2023.

Popis cjelina

[Opis data set-a 3](#_Toc127214065)

[Postavljena pitanja 4](#_Toc127214066)

[Rješenje 5](#_Toc127214067)

[Zadatak (1) 5](#_Toc127214068)

[Dohvaćanje potrebnih vrijednosti 5](#_Toc127214069)

[Definiranje grafa G 7](#_Toc127214070)

[Crtanje grafa 8](#_Toc127214071)

[Pod grafovi 10](#_Toc127214072)

[Zadatak (2) 29](#_Toc127214073)

[Izlazni stupanj, ulazni stupanj i distribucija stupnjeva 29](#_Toc127214074)

[Jako povezane komponente, dijametar, radijus 31](#_Toc127214075)

[Zadatak (3) 32](#_Toc127214076)

[Centralnost međupoloženosti 32](#_Toc127214077)

# **Opis data set-a**

U ovom radu analizirati ću problem kompleksnih mreža u domeni transporta. Data set se sastoji od dvije radne stranice nodelist i edgelist.

Nodelist (sadrži 125 luka) sadrži stupce:

* **ID** – identifikacija grada
* **LABEL** – naziv grada
* **LAT** – zemljopisna širina (u stupnjevima)
* **LNG** – zemljopisna dužima (u stupnjevima)

Edgelist predstavlja broj odrađenih vožnji između dva odredišta te se sastoji od sljedećih stupaca:

* **source** – početni grad
* **target** – krajnji grad
* **weight** – broj vožnji

U ovom radu konstruirati ću težinski usmjereni graf. Dostupne informacije poput *usmjerenosti bridova* korištenjem *polazišnog vrha* iz varijable source te *odredišnog vrha* iz varijable target i *težine bridova* iz varijable weight potaknuli su me na tu odluku.

# **Postavljena pitanja**

1) (SREDIŠNJA RAZINA)

Trguju li gradovi sa susjednim gradovima,

dali su te trgovačke rute formirale zajednice?

2) (GLOBALNA RAZINA)

Dali je promet prema gradovima veći od prometa iz gradova?

Kakva je distribucija prometa po zajednicama?

3) (LOKALNA RAZINA)

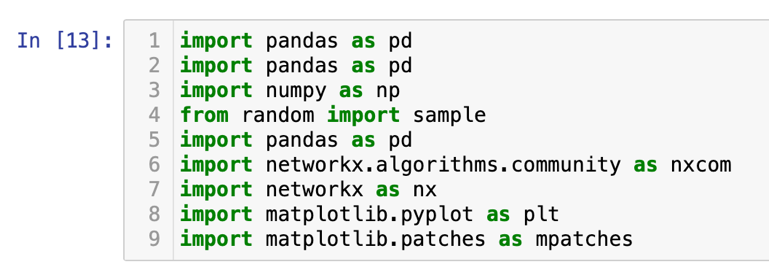
Gradove povezuju rute koje prenose teret te je logično da će neki gradovi imati mali broj ruta. Postoji li opasnost od ne mogućnosti transporta ukoliko se prekine ruta? Koji gradovi su u opasnosti od ovog problema tj. koji gradovi bi bili odcjepljeni od ostalih gradova?

# **Rješenje**

## Zadatak (1)

### Dohvaćanje potrebnih vrijednosti

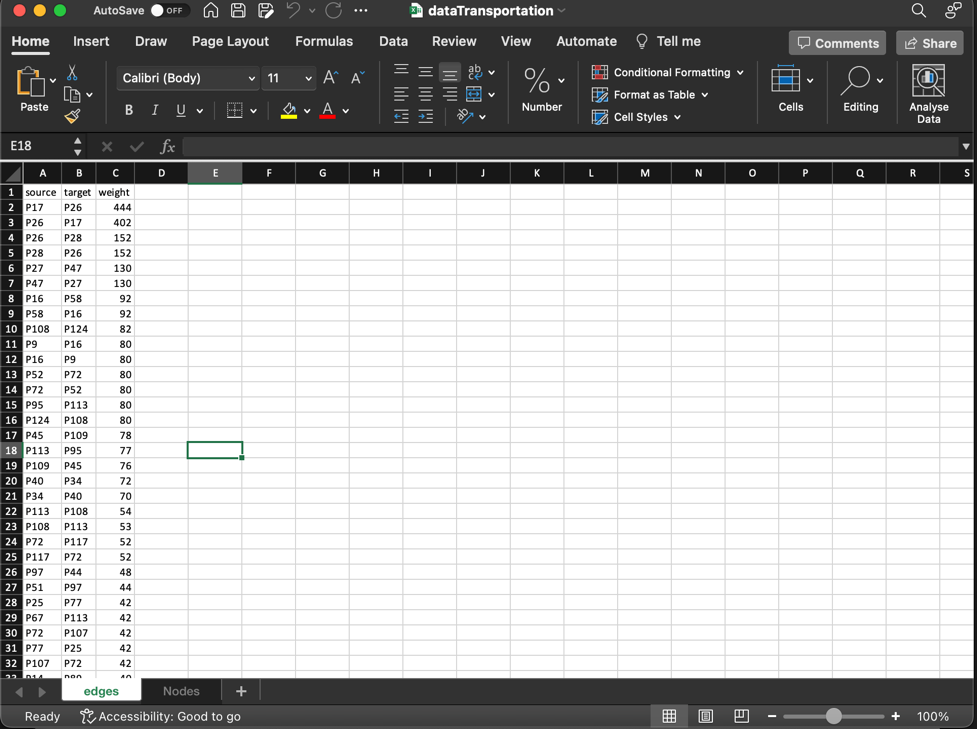
Uvoz potrebnih biblioteka.



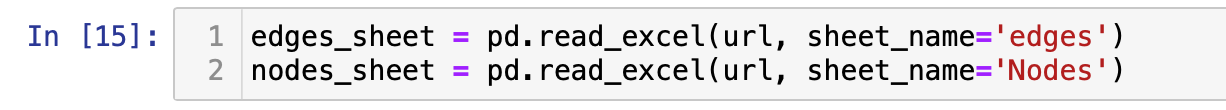
Excel izvor podataka postavio sam na svoj GitHub ( radi lakšeg izvođenja te testiranja koda ) .



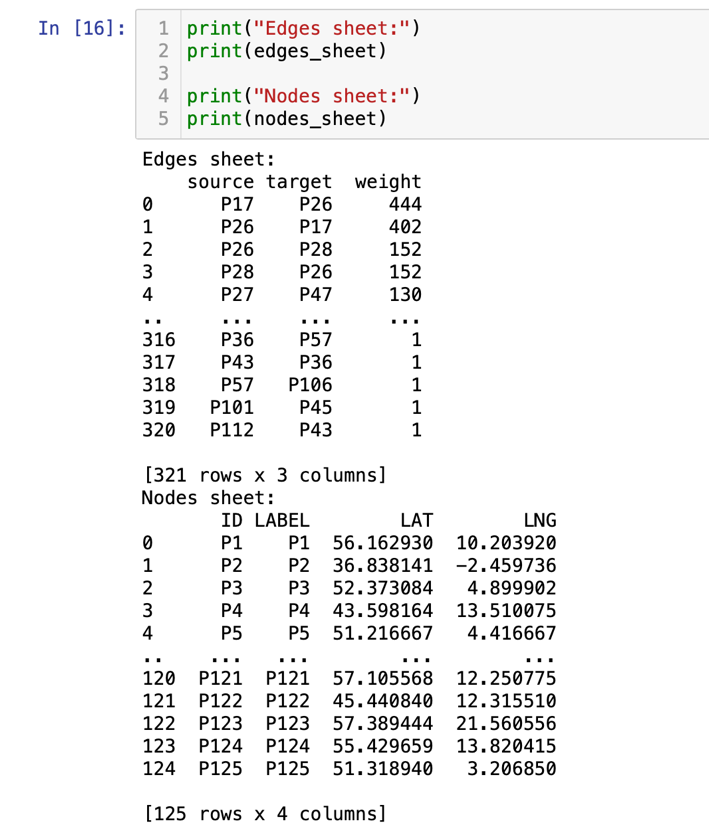
Pristupom na link te pohranom datoteke dataTransportation.xlsx možemo vidjeti stvarni oblik podataka u programu Excel.



Zanimljiv je oblik bilježnice (dvije radne stranice u donjem lijevom kutu). Njima možemo pristupiti pomoću slijedećih naredba te podijeliti data set u popis bridova i vrhova.

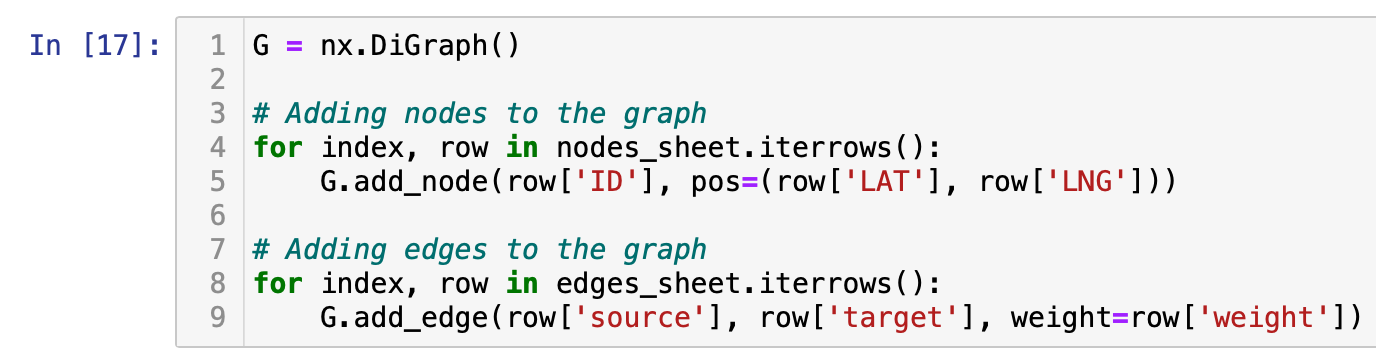


Možemo i ispisati pohranjene vrijednosti vrhova i bridova.

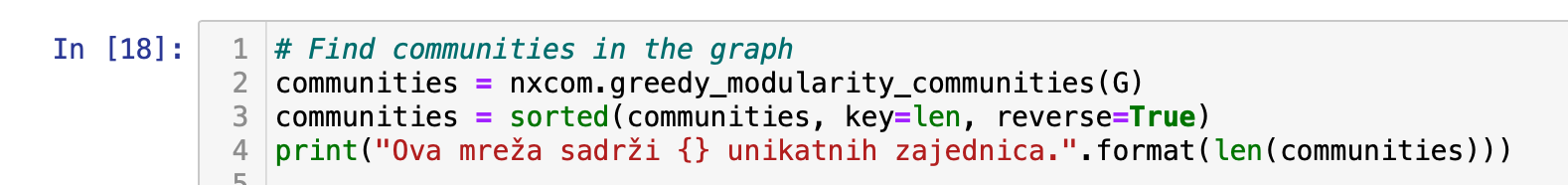


### Definiranje grafa G

Nakon uspješno dohvaćenih vrhova i bridova možemo kreirati usmjereni težinski graf. Graf ćemo postaviti na usmjereni radi dostupnih podataka o bridovima tj. stupci source i target. Stupac source početak je brida dok u stupcu target nalazi se kraj brida (u točki), samim tim možemo odrediti smjer brida. Graf će biti težinski radi dostupnog stupca weight koji definira popularnost puta tj. broj korištenja puta. Sada možemo dodati vrhove te odgovarajuće bridove u graf G.

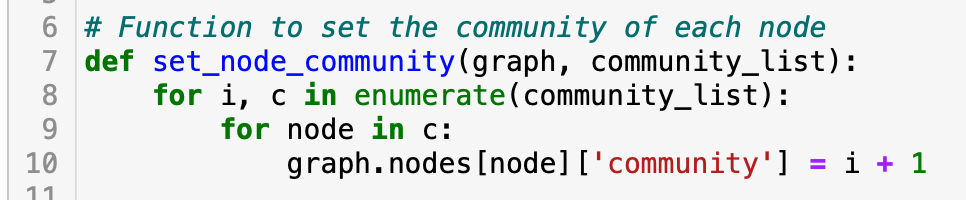


Korištenjem pohlepnog algoritma kreirati ćemo zajednice u grafu.

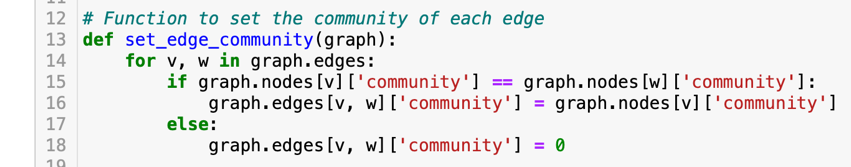


Izvođenjem koda vidljivo je da pohlepni algoritam kreira 14 zajednica. Želim naglasiti da će zajednice biti predstavljene vrijednostima od 0 do 13 uvećanjem od 1.

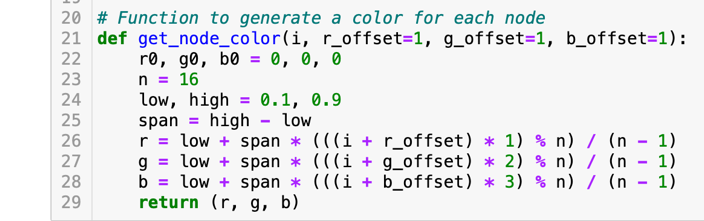
Slijedeća funkcija dodjeljuje zajednicu svakom vrhu.



Slijedeća funkcija dodjeljuje zajednicu svakom bridu.



Slijedeća funkcija dodjeljuje RGB vrijednost za svaku zajednicu ( generira se vrijednost pomoću postavljene formule prvo RED vrijednost piksela potom GREEN vrijednost piksela te naposlijetku BLUE vrijednost piksela) ova funkcija je odlična jer nam omogućuje lako skaliranje boja zajednica tj. dodavanjem novih zajednica automatski se generira unikatna boja zajednice.



Prikaz izlaza funkcije get\_node\_color:



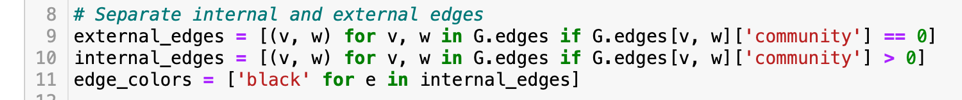
Prethodne tri funkcije dostupne su javno na projektu *analiza mreže karate kluba*.

Sada možemo pozvati prethodno definirane funkcije.

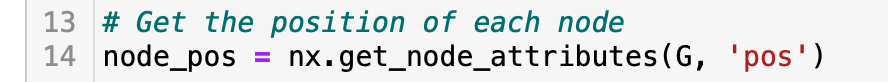
Text

Description automatically generated with medium confidence

Dodatno sam definirao bridove koji se nalaze unutar zajednice internal\_edges te bridove koji povezuju zajednice external\_edges. Boja bridova koji se nalaze u zajednici postavljena je na crnu dok je boja bridova koji povezuju zajednice siva.



Svakom vrhu sada možemo dati poziciju LAN i LNG.



### Crtanje grafa

Sada možemo nacrtati mrežu.

Graphical user interface, text, application

Description automatically generated

Mreži ćemo dati veće dimenzije od 100x100. Za crtanje grafa potrebno je dostaviti slijedeće parametre u naredbu nx.draw\_networkx:

1. **G** – graf
2. **pos** – pozicija točaka (iz varijable node\_pos koja čita graf G u kojem su pohranjene koordinate)
3. **node\_size** – veličina točaka pri crtanju
4. **node\_color** – boja točaka (grupirali smo ih preko funkcije te dinamički dodijelili boju)
5. **edgelist** – popis bridova za crtanje (ovdje sam postavio samo bridove koji se nalaze u zajednicama)
6. **edge\_color** – boja bridova (dobivamo iz varijable edge\_colors)

Rječnik nizic definiran je u svrhu identifikacije vrhova po zajednici te za crtanje legende u grafu.

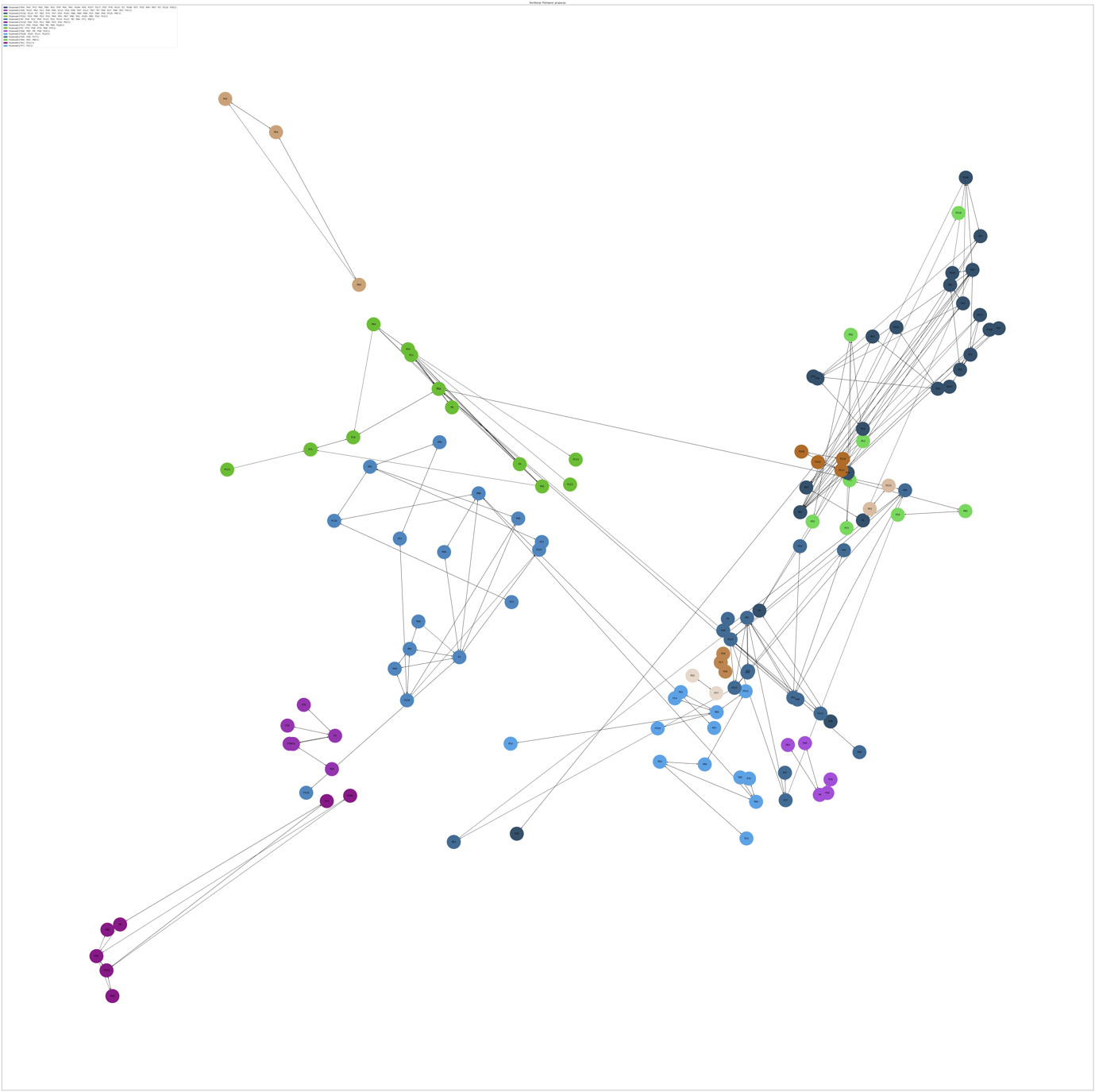
Text

Description automatically generated

U legendi grafa direktno sam upario dinamično generirane boje te generirane zajednice.

Micanje frozenset-a moguće je no ovakav prikaz sortiran je po silaznom principu (najveća zajednica prema najmanjoj zajednici) indeksi zajednica od najveće (zajednica[0]) do najmanje (zajednica[13]).

Prikaz grafa sa legendom i naslovom:



Radi velikih dimenzija koje su stvarno potrebne kako bi se graf pravilno vidio legenda je izuzetno mala. Dimenzije nam omogućuju zumiranje do 100X samim tim legenda će se vidjeti perfektno ukoliko se plot preuzme lokalno.

### Pod grafovi

Pohlepni algoritam kreira 14 unikatnih zajednica (indeksi od 0 do 13). Moj cilj bio je svaki od tih grafova izvući te nacrtati na zasebnom ekranu. Za ovaj zadatak vratio sam se na početak te definirao stupac u data set-u vrhova i bridova pod nazivom group. Moja zamisao je definirati dinamički grupe tj. zajednice pridruživanjem vrijednosti iz varijable communities i ID-a svakog vrha:

Text

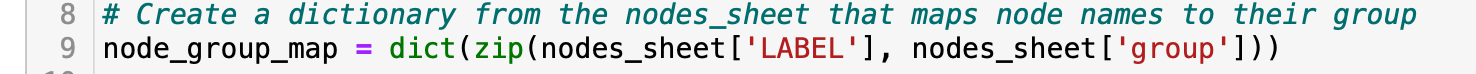
Description automatically generated

Pohranjene vrijednosti mogu se pročitati iz rječnika node\_group\_map oblika tuple (prikaz prvih dvanaest točaka):

Table

Description automatically generated

Naredba za generiranje node\_group\_map rječnika:



Potrebno je kreirati funkciju koja će pročitati vrijednost brida kojeg želimo usporediti.

Za usporedbu koristiti ću svoj algoritam koji uspoređuje vrhove u zajednici. Kreirati ću dva stupca u data set-u bridova source\_group i target\_group. Logički ću gledati dali brid dolazi 'izvana' (brid pokušava spojiti dvije zajednice) te ću takve bridove maknuti, a bridovi koji se nalaze u promatranoj zajednici imati će podrijetlo tj. source\_group i target\_group iz iste zajednice.

Text

Description automatically generated

Rezultat sam prvo pohranio u CSV format radi lakše analize, a potom provjere i analize aktivno ga koristim za kreiranje pod grafova.

Prikaz bridova grupiranih u zajednice u CSV formatu:

Table

Description automatically generated

Sada možemo pozvati i prikazati pod grafove:

Graphical user interface, text, application

Description automatically generated

Grafove pozivamo putem for funkcije sa parametrom z. Varijable edge\_0 i node\_0 dinamički mijenjaju svoj stanje po iteraciji. Ranije sam spomenuo indexe od najvećeg pod grafa do najmanjeg i upravo taj pristup koriste naredbe. Brzi pregled koda:

* Linija 2 i 3: Definiraju bridove i vrhove pod grafa ovisno o parametru ( na prvom pokretanju najveći pod graf)
* Linija 5: Kreira novi graf koji ću koristiti za definiranje pod grafova
* Linija 8 i 9: Čitaju vrhove pod grafa te njihovu lokaciju na ekranu kako bi dosljednost s originalnog grafa bila održana
* Linija 12 i 13: Čitaju bridove pod grafa njihovu početnu i krajnju točku te težinu
* Linija 15: Poziva atribute vrhova
* Linija 18 do 21: Nemaju pretjeranu važnost (postavljene su radi izuzetno velike količine podataka kao logička provjera tj. dali se grupiraju dobri vrhovi i bridovi)
* Linija 24: Omogućuje veliki prikaz grafa
* Linija 25 do 27: Uparuje ime zajednice pod grafa i pripadajućih vrhova **sa istom bojom** koju je imao u prethodnom grafu
* Linija 29 do 32: Crta graf

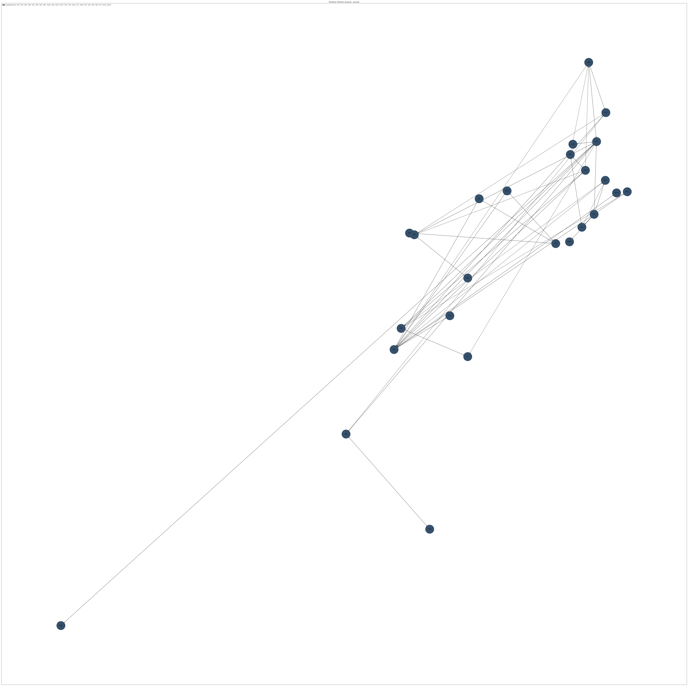
Ovi koraci ponavljaju se za sve pod grafove!

Prikazati ću svih 14 pod grafova sa dodatnom analizom i ispisom njihovih bridova i vrhova:

1. Pod graf [0] - najveći

Table

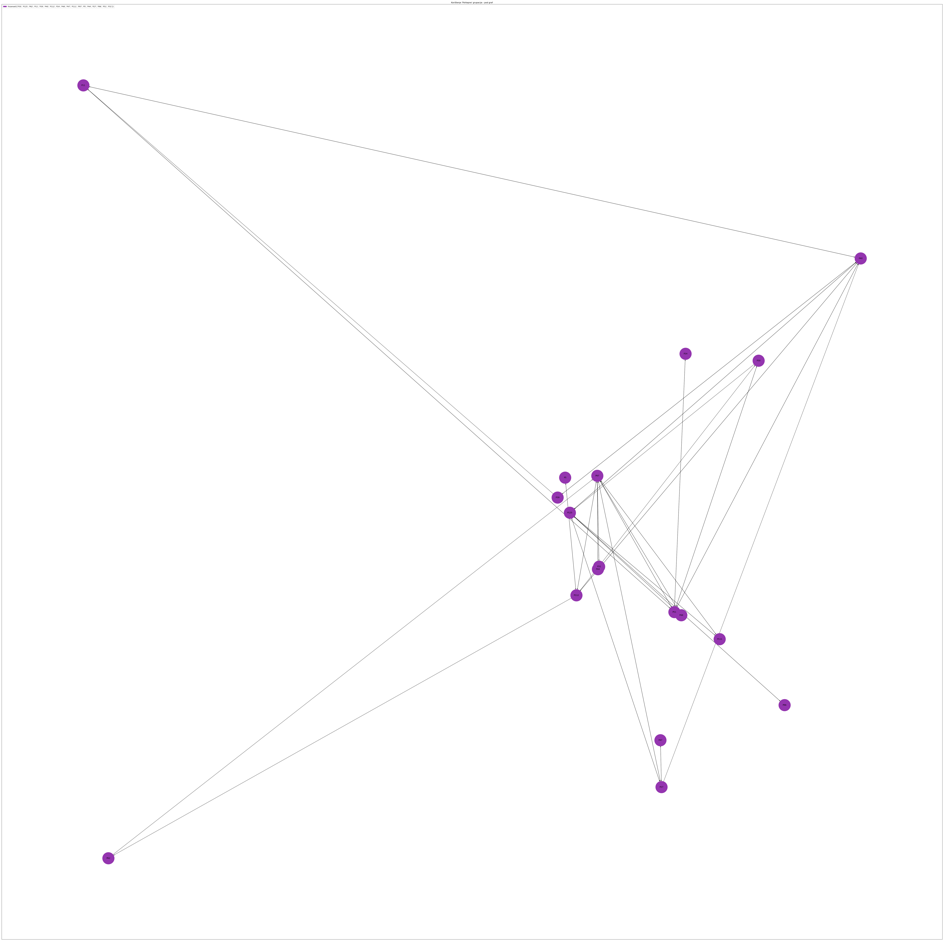
Description automatically generated



1. Pod graf [1]

Table

Description automatically generated



1. Pod graf [2]

Table

Description automatically generated

A picture containing sky, outdoor, day, several

Description automatically generated

1. Pod graf [3]

Table

Description automatically generated

Chart, line chart

Description automatically generated

1. Pod graf [4]

Table

Description automatically generated

Chart

Description automatically generated

1. Pod graf [5]

Table

Description automatically generated

Chart, line chart

Description automatically generated

1. Pod graf [6]

Table

Description automatically generated

Chart

Description automatically generated with medium confidence

1. Pod graf [7]

Table

Description automatically generated

A picture containing text, outdoor, line, day

Description automatically generated

1. Pod graf [8]

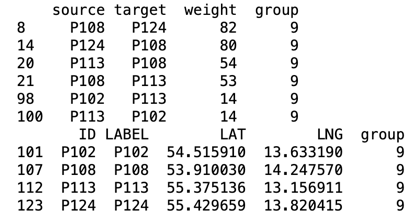
Table

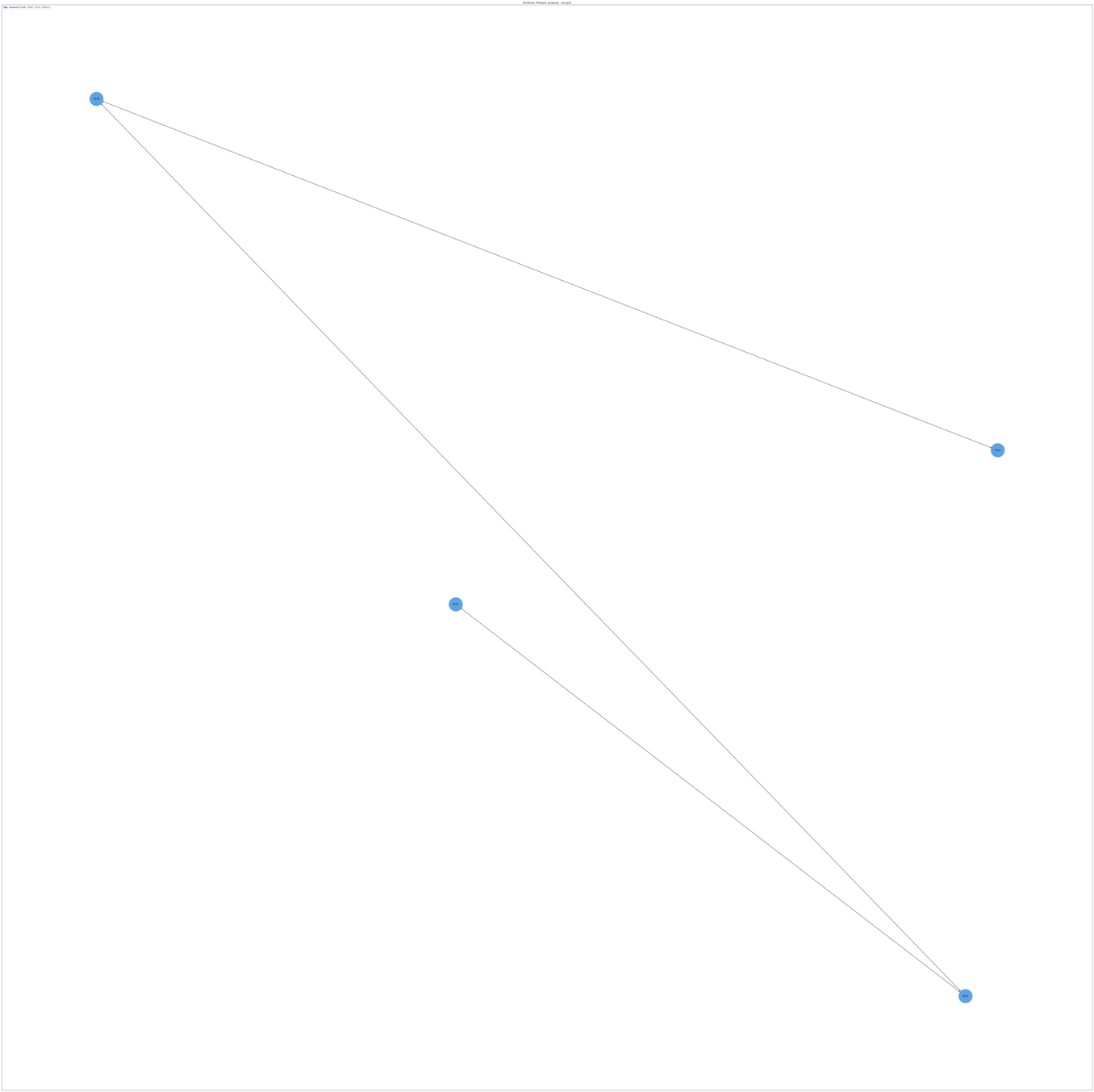
Description automatically generated with low confidence

A picture containing kite, colorful, line, day

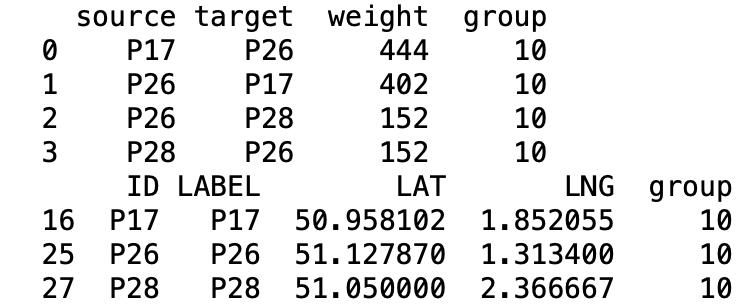
Description automatically generated

1. Pod graf [9]





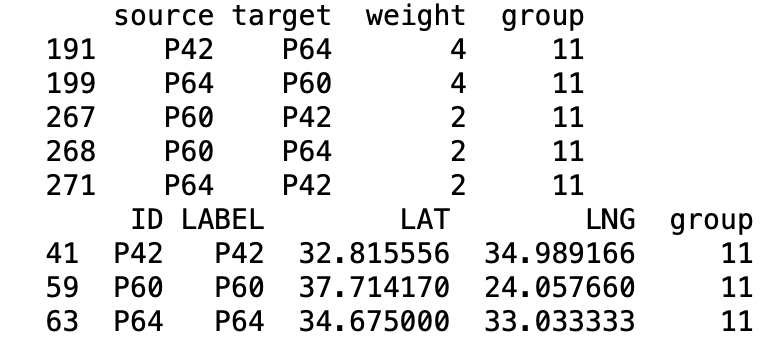
1. Pod graf [10] – anomalija u apliciranju boja. Veze i bridovi dobri



Polygon

Description automatically generated

1. Pod graf [11] – anomalija u apliciranju boja. Veze i bridovi dobri



Chart, line chart

Description automatically generated

1. Pod graf [12]

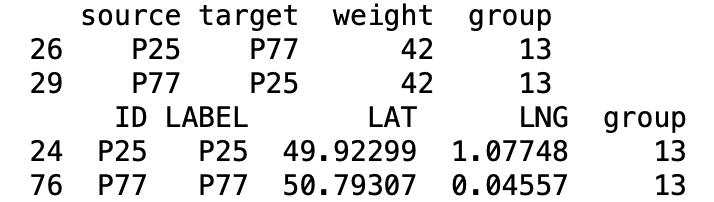
Text

Description automatically generated

Line chart, polygon

Description automatically generated

1. Pod graf [13] – najmanji pod graf

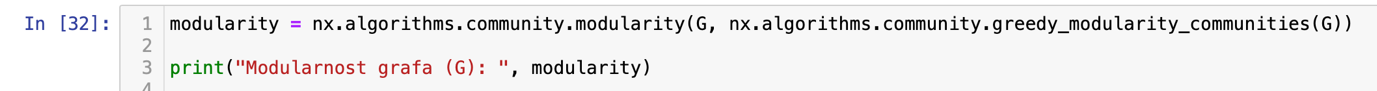


Chart, line chart, polygon

Description automatically generated

Dodatno sam napravio analizu grafa G:

* **Modularnost** - Vrijednost modularnosti kreće se od -1 do 1, s vrijednostima blizu 1 koje označavaju dobro grupiranje ili strukturu zajednice na grafikonu, a vrijednosti blizu -1 ukazuju na loše klasteriranje. Vrijednost modularnosti od približno 0,76 je dobar rezultat koji ukazuje na da na grafikonu postoji snažna struktura zajednice te odgovara na postavljeno pitanje ovog zadatka.





* **Ispis bridova i vrhova po zajednici** – varijabla vraća broj bridova i vrhova po zajednici uz cilj dodatne provjere pod grafova.

Text

Description automatically generated

Table

Description automatically generated with low confidence Text

Description automatically generated

* **Omjer bridova u zajednici i izvan zajednice** - Vrijednost omjera bridova unutar zajednica prema bridovima između zajednica pokazatelj je koliko su dobro definirane zajednice na grafikonu. Visoka vrijednost za ovaj omjer znači da postoji mnogo više rubova unutar zajednica u usporedbi između zajednica, što sugerira da su zajednice dobro definirane i različite jedna od druge.

U mom slučaju, vrijednost omjera je približno 6,47 te ta vrijednost pokazuje da postoji mnogo više bridova unutar zajednica nego između zajednica. Ovaj proces dodatna je potvrda da su zajednice jedinstvene i dobro podijeljene.

Text

Description automatically generated with low confidence



## Zadatak (2)

### Izlazni stupanj, ulazni stupanj i distribucija stupnjeva

Stvorio sam tri stupčasta grafikona pomoću biblioteke Plotly Express.

Plotly Express biblioteka omogućuje lako i uredno crtanje grafova sa unaprijed definiranim korisnim funkcijama poput povećanja i smanjenja segmenata, suženja promatranog intervala te legende prilikom prelaska kursorom preko grafa.

Prvi grafikon prikazuje distribuciju unutarnjih stupnjeva čvorova na grafu G, drugi grafikon prikazuje distribuciju vanjskih stupnjeva čvorova na grafu, a treći grafikon prikazuje distribuciju ukupnih stupnjeva (zbroj in-stupnjeva i out-degree) čvorova u grafu.

Graphical user interface, text, application

Description automatically generated

Prikazane su vizualizacije: izlazni stupanj, ulazni stupanj i distribucija stupnjeva.

Graphical user interface, chart, application

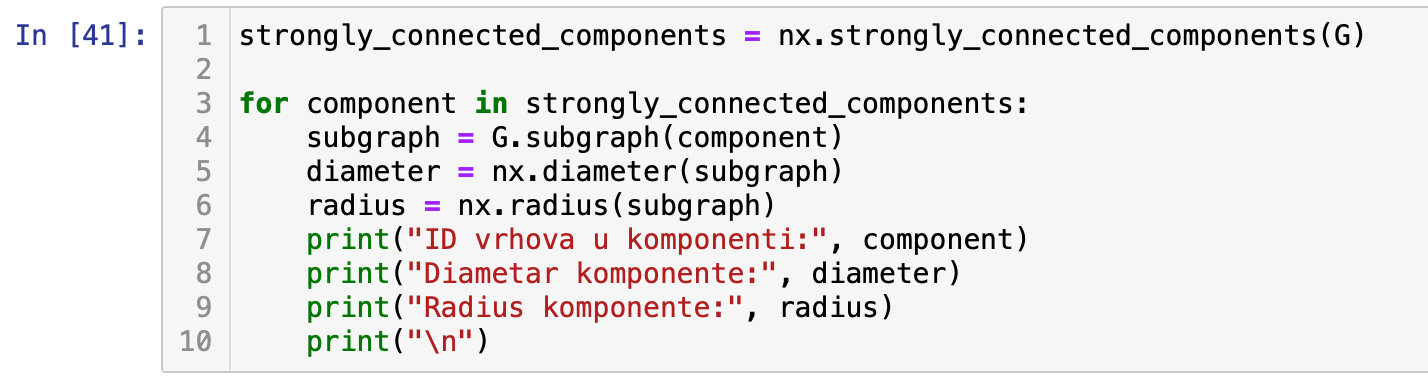
Description automatically generated

**Možemo opaziti da su ulazni i izlazni stupnjevi vrhova jednaki. Najčešće ulazne i izlazne veze grafova su 2 i 3.**

### Jako povezane komponente, dijametar, radijus

Izračunao sam jako povezane komponente usmjerenog grafa G pomoću funkcije nx.strongly\_connected\_components, koja vraća popis skupova, gdje svaki skup predstavlja jako povezanu komponentu u grafu.

Dodatno sam izračunao dijametar i radijus.



Text

Description automatically generated

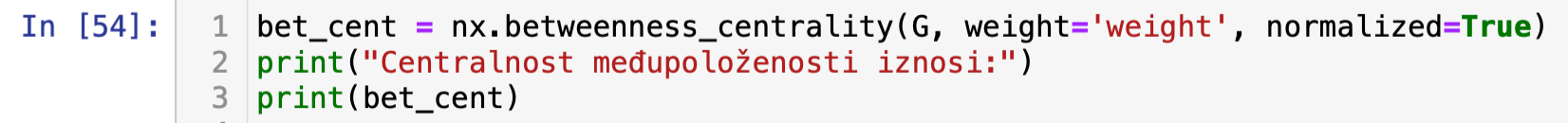
Text

Description automatically generated

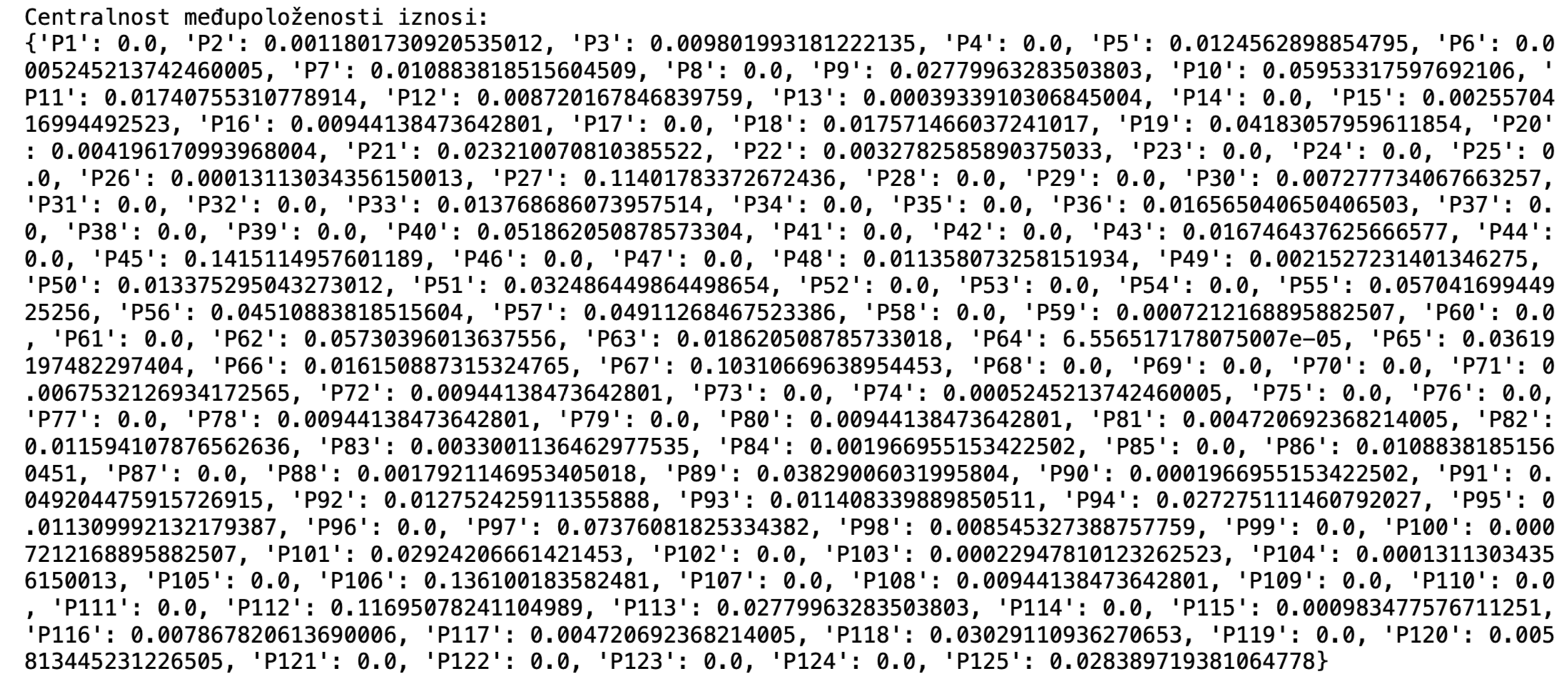
## Zadatak (3)

### Centralnost međupoloženosti

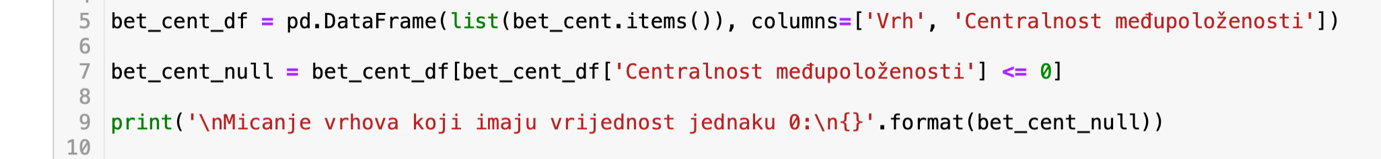
Izračunao sam centralnost međuposloženosti grafa kako bi utvrdio potencijalne slabosti tj. gradove koji bi mogli postati izolirani. Prvo sam izračunao međuposloženosti pomoću naredbe nx.betweenness\_centrality koja sadrži vrhove i bridove grafa G te uzima u obzir karakteristike grafa poput usmjerenosti i težinskih vrijednosti bridova. Podaci su potom i normalizirani.



Nakon pokretanja vidljiva je lista u obliku tuple tj. **{[oznaka\_node-a]:[centralnost međuposloženosti za pripadajući node]}**.



Vrhovi s malim vrijednostima imaju izrazito veliku šansu (pri micanju bridova grafa) ostati nepovezani. Zbog ljepše vizualizacije izbacio sam sve vrhove s vrijednosti 0 (to su najosjetljiviji gradovi te sam ih prikazao u slijedećem ispisu).

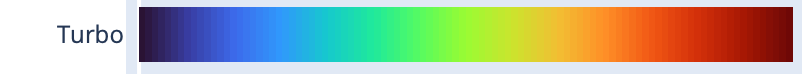


Table

Description automatically generated

Ostale vrhove sam prikazao pomoću plotly.expres:

* Dodatno skalirana os X (oznake su bolje vidljive)
* Gradient boja označuje centralnost međuposloženosti čvora



* Dodano dinamičko alociranje vrijednosti iz pandas data set-a (prelazak kursora preko grafa)

Chart, bar chart

Description automatically generated

* Mogućnost izoliranja regija

Chart

Description automatically generated

**Ova vizualizacija prikazuje vrhove (plavih boja) koji pri micanju vrlo malo čvorova postaju odvojene komponente, a kao takvi nemaju mogućnost komunicirati s ostatkom grafa.  
Vrhovi: P45, P106, P112, P27 i P67 najotporniji su na promjene tj. micanje bridova u grafu.**