

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
Univerzitetni študijski program
Finančna matematika

FINANČNI PRAKTIKUM
PORAZDELITEV RAZDALJ V NEUTEŽENIH
GRAFIH

AVTORJA: Jaka Svetek, Tajda Koščak

Ljubljana, 2022

1 Navodilo

Naj bo G neutežen graf. Porazdelitev razdalj grafa G , nam pove za vsako število i , število parov vozlišč, ki so oddaljeni za i . Preuči in analiziraj porazdelitev razdalj za enostavne družine grafov (drevesa, cikli, cikli z nekaj dodanimi povezavami, unije ciklov).

2 Potek dela

Vhodni podatek najinega problema je neuteženi graf $G = (V, E)$. Problema bi se lahko lotila v Pythonu, a sva po predlogu profesorja raje uporabila SageMath, saj že vsebuje mnogo funkcij na grafih. Najprej sva zapisala funkcijo, ki nariše graf porazdelitve razdalj za en sam graf $G = (V, E)$, nato pa sva jo uporabila v najni glavni funkciji, ki sprejme poljubno mnogo grafov in vrne en graf, na katerem so prikazane porazdelitve razdalj vseh sprejetih grafov. Ta način prikaza sva uporabila zaradi boljše preglednosti in primerjave med družinami grafov. Nato sva se lotila analize in testirala različne vrste grafov. Grafe sva generirala naključno z že vgrajenimi funkcijami v SageMath-u.

2.1 Opis funkcij

Osnovna funkcija najine projektne naloge, je funkcija NarisiGraf, ki sprejme poljuben graf in nariše njegovo porazdelitev razdalj. Graf nam na abscisni osi pove razdaljo, na ordinatni osi pa števila parov vozlišč, oddaljena za to razdaljo. Koda funkcije je napisana spodaj.

```
def NarisiGraf(graf):
    razdalje = graf.distance_all_pairs()
    l = razdalje.values()
    k = []
    for i in l:
        for j in i.values():
            k = k + [j]
    pojavitve = {}
    for i in k:
        if i in pojavitve:
            pojavitve[i] += 1
        else:
            pojavitve[i] = 1
    lists = sorted(pojavitve.items())
```

```

x, y = zip(*lists)
r = list(y)
for i in range(len(r)):
    r[i] = r[i]/2
y = tuple(r)
y = y[1:len(y)]
x = x[1:len(x)]
plt.plot(x, y)

```

To funkcijo sva nato uporabila v večji funkciji, ki sprejme niz grafov in vse porazdelitve razdalj le teh nariše na en graf.

```

def PorazdelitevRazdalj(grafi):
    for graf in grafi:
        NarisiGraf(graf)

```

3 Rezultati

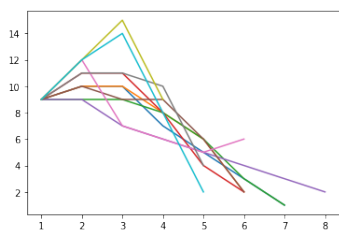
Najprej sva s pomočjo narisanih grafov analizirala drevesa, nato cikle, cikle z dodanimi povezavami in pa unije ciklov. Grafi porazdelitev razdalj prikazujejo število parov vozlišč oddaljenih za x v odvisnosti od razdalje x (torej so na abscisni osi razdalje, na ordinatni osi pa števila parov vozlišč, oddaljena za to razdaljo).

3.1 Drevesa

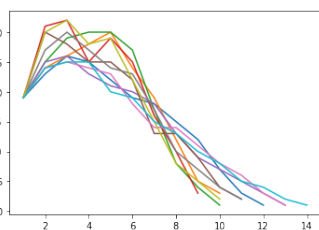
Zgenerirala sva 10 naključnih dreves na desetih točkah, na dvajsetih vozliščih in pa še dvajset dreves na petdesetih vozliščih. Funkcija za generiranje naključnih dreves je v SageMath že vgrajena, kot argument sprejme število vozlišč. Prikazana je spodaj.

```
T = graphs.RandomTree(n)
```

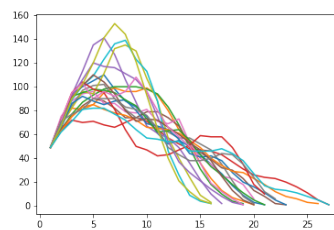
Iz narisanih grafov opaziva, da število parov najprej narašča z medsebojno razdaljo, nato pa strmo pada. Po najinih pričakovanjih je število parov, oddaljenih oddaljenih za 1, ravno $n-1$, saj je v drevesih z n vozlišči, točno $n-1$ povezav. Najdaljša pot v drevesu je enaka desnemu krajišču definicijskega območja porazdelitve. Večje, kot je definicijsko območje porazdelitve razdalj grafa, manj razvejano je drevo.



Slika 1: 10 vozlišč



Slika 2: 20 vozlišč

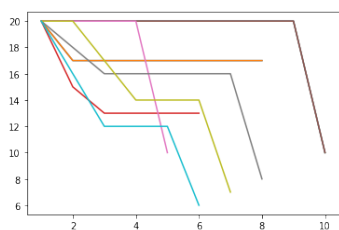


Slika 3: 50 vozlišč

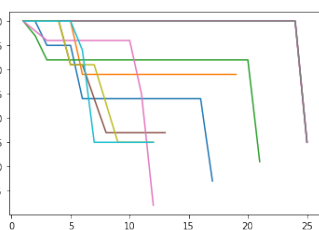
3.2 Cikli

Zgenerirala sva 10 naključnih ciklov na dvajsetih točkah, na petdesetih vozliščih in pa še dvajset ciklov na sto vozliščih. Funkcijo za generiranje ciklov dobimo v SageMath-u tako, da vzamemo funkcijo za generiranje naključnih grafov, a ji kot prvi argument (stopnja vseh vozlišč) podamo 2 (saj imajo vsa vozlišča v ciklih stopnjo 2), kot drugi argument pa sprejme število vozlišč. Prikazana je spodaj.

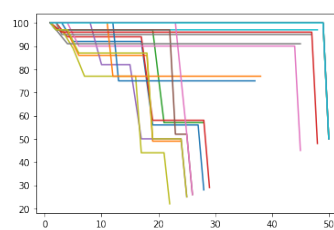
```
C = graphs.RandomRegular(2, n)
```



Slika 4: 20 vozlišč



Slika 5: 50 vozlišč

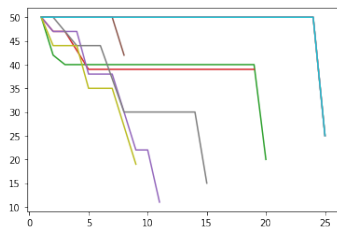


Slika 6: 100 vozlišč

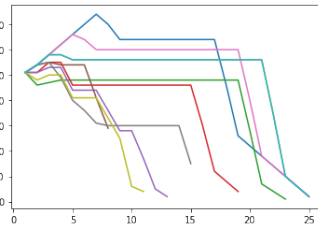
Opaziva, da so vsi grafi padajoči in pa odsekoma konstantni. Iz vsakega grafa porazdelitev razdalj ciklov lahko razberemo iz koliko nepovezanih delov je (koliko nepovezanih ciklov ga sestavlja) in pa velikost največjega cikla. Število nepovezanih delov je enako številu strogo padajočih odsekov grafa porazdelitev razdalj $+1$. Velikost največjega cikla v grafu je enaka desnemu krajišču definicijskega območja porazdelitve.

3.3 Cikli z dodanimi povezavami

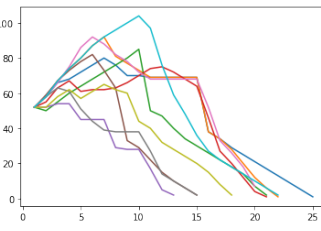
Zgenerirala sva 10 naključnih ciklov na petdesetih vozliščih. Dodala sva jim prvo eno povezavo, nato pa še eno. Funkcija za dodajanje naključnih povezav v graf G je v SageMath-u že vgrajena. Prikazana je spodaj.



Slika 7: Cikli



Slika 8: 1 bližnjica



Slika 9: 2 bližnjici

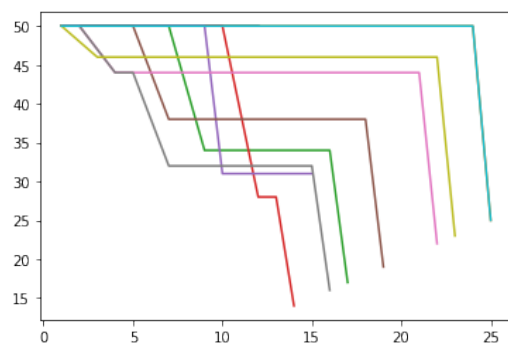
```
G.add_edge(G.complement().random_edge())
```

Iz zgornjih grafov po pričakovanjih opaziva, da se na desni strani grafi začnejo hitreje bližati, saj se z dodajanjem povezav ciklom razdalje med vozlišči kvečjemu zmanjšujejo. Več vozlišč kot bi dodala, bolj bi se maksimum funkcij pomaknil v levo. Če bi dodala vse možne povezave (graf postane poln), bi bila največja razdalja med vozliščema enaka 1, torej bi bil graf porazdelitev razdalj le ena točka in sicer $(1, \frac{n(n-1)}{2})$.

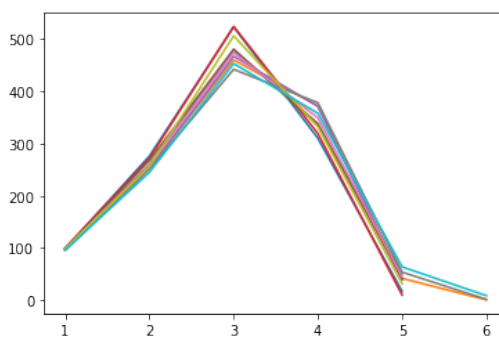
3.4 Unije ciklov

Zgenerirala sva 10 naključnih ciklov na petdesetih vozliščih. Nato sva naredila 10 različnih unij parov teh ciklov. Funkcija za generiranje unije grafov G in H je v SageMath-u že vgrajena. Prikazana je spodaj.

```
U = G.union(H)
```



Slika 10: Cikli



Slika 11: Unije parov ciklov

Opaziva, da se unije zelo prilegajo, kar je logično, saj sva dejansko grafom le dodala ogromno povezav in se kot že v točki 3.3 dafinicijsko območje grafa manjša- graf porazdelitve razdalj se viša in približuje y osi.