

# Socijalne mreže

Praktični projekat

-Izveštaj-

## Učitavanje grafova:

Nakon pokretanja programa, korisnik treba da unese koji graf želi da učitá, odnosno nad kojim grafom želi analizu. Na osnovu izbora korisnika, poziva se jedna od metoda za učitavanje grafova.

### Konvertovanje grafa:

Svaka metoda za učitavanje grafova kao povratnu vrednost vraća neusmeren graf tako što pozove metodu *transformFromDirToUndir(DiGraph g)*.

Ideja je da prolazimo kroz učitán usmeren graf i proveravamo da li u njemu postoje čvorovi koji su povezani linkom dva puta i ukoliko je bar jedan od linkova negativan, u novom grafu link ima negativnu vrednost. Inače, link dobija pozitivnu vrednost.

```
def transformFromDirToUndir(g1):  
    undir = nx.Graph()  
    undir.add_edges_from(g1.edges(), affinity="")  
    for u, v, d in g1.edges(data=True):  
        af1 = g1[u][v]['affinity']  
        af2 = ""  
        if(v, u) in g1.edges:  
            af2 = g1[v][u]['affinity']  
        if af1 == "negative" or af2 == "negative":  
            undir[u][v]['affinity'] = "negative"  
        else:  
            undir[u][v]['affinity'] = "positive"  
    return undir
```

## Provera klasterabilnosti grafova:

Identifikovanje komponenti grafa se realizuje koristeći BFS algoritam.

Modifikacija algoritma se ogleda u tome što jednu komponentu treba da čine čvorovi koji su međusobno povezani linkovima koji imaju pozitivnu vrednost, ukoliko su povezani negativnim linkom, ne mogu biti u istoj komponenti. Zbog efikasnosti ne brišemo negativne linkove, već ih zanemarujemo.

```
def bfs(visited, g, node):
    comp = set()
    queue = list()
    comp.add(node)
    visited.add(node)
    queue.append(node)

    while queue:
        current = queue.pop(0)
        neighbours = list(nx.neighbors(g, current))

        for neighbour in neighbours:
            a = g.get_edge_data(current, neighbour)

            if a['affinity'] == 'negative':
                continue

            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)
                comp.add(neighbour)

    return frozenset(comp)
```

```
def identifyComponents(g):
    components = set()
    visited = set()

    for node in g:
        if node not in visited:
            components.add(bfs(visited, g, node))
    numberOfC = str(len(components))
    print("Graf sadrzi " + numberOfC + " klastera.")
    return frozenset(components)
```

Metoda koja kreira skup grafova od skupa klastera funkcioniše tako što za svaki klaster, koji je predstavljen kao skup, prolazi kroz njegove čvorove i za svaki čvor proverava da li u njegovoj listi susedstva postoji neki čvor, ukoliko postoji u kreirani graf dodaje ta dva čvora i link koji ima vrednost iz originalnog grafa.

```
def makeClusterNetwork(clusters, g):
    setOfClusters = set()
    i = 0

    for cluster in clusters:
        graph = nx.Graph()
        for node in cluster:
            graph.add_node(node)
            neighbours = list(nx.neighbors(g, node))
            for node2 in cluster:
                if node == node2:
                    continue
                if node2 in neighbours:
                    graph.add_edge(node, node2, affinity = g.get_edge_data(node, node2))
            setOfClusters.add(graph)
        i = i + 1
    print("Napravljeno je " + str(i) + " grafova klastera.")
    return setOfClusters
```

Metoda koja proverava klasterabilnost grafa kao parametar prima skup grafova i originalni graf. Implementirana je tako da za svaki graf iz skupa klastera koji ima više od dva čvora (ukoliko ima jedan ili dva čvora sigurno ispunjava uslov klasterabilnosti), proverava da li unutar tog grafa postoji link sa negativnom vrednošću. Ukoliko postoji, taj graf nije klasterabilan i pamtimo link koji narušava klasterabilnost kako bismo ih isporučili korisniku. Ovakav graf dodajemo u listu antikoalicija, a ukoliko nismo pronašli negativan link, onda dodajemo u listu koalicija. Ove dve liste su povratna vrednost metode.

```

def isClusterable(clusters, g):
    coalitions = list()
    antiCoalitions = list()
    negativeEdges = list()
    ret = list()
    br = 0
    for c in clusters:
        if len(c) > 2:
            n = [(u,v) for (u,v,d) in c.edges(data=True) if c[u][v]['affinity']['affinity'] == 'negative']

            if len(n) == 0:
                coalitions.append(c.nodes)
            else:
                antiCoalitions.append(c.nodes)
                negativeEdges.extend(n)
        else:
            coalitions.append(c.nodes)
    if len(antiCoalitions) > 0:
        print("Ovaj graf nije klasterabilan.")
        print("Postoji ", len(negativeEdges), " grana koje narušavaju klasterabilnost.")
        x = input("Da li želite da se prikazu? (y/n)")
        if(x == 'y'):
            print("Treba ukloniti grane:")
            for edge in negativeEdges:
                print(edge)
            print("Prikazane su grane koje narušavaju klasterabilnost grafa (", len(negativeEdges), ").")
    else:
        print("Ovaj graf je klasterabilan.")
    ret.append(coalitions)
    ret.append(antiCoalitions)
    return ret

```

Metoda koja kreira mrežu klastera kao parametre prima skup klastera i originalni graf. Implementirana je tako da prolazi kroz sve klastere koji su sada predstavljeni kao čvorovi i dodaje ih u kreirani graf. Nakon toga proverava za svaka dva klastera da li su povezani, ukoliko jesu, dodaje link sa negativnom vrednošću.

```

def toNetworkClusters(clusters, g):
    print("Kreiranje mreže klastera...")
    clusterNetwork = nx.Graph()
    count = 1
    for cluster in clusters:
        cluster.graph['name'] = str(count)
        count = count + 1
        clusterNetwork.add_node(cluster)

    for cluster1 in clusters:
        for cluster2 in clusters:
            if cluster1 == cluster2:
                continue
            neighbour = False

            for node1 in cluster1:
                for node2 in cluster2:
                    if node1 in g.neighbors(node2):
                        neighbour = True
            if neighbour:
                clusterNetwork.add_edge(cluster1, cluster2)
    nx.set_edge_attributes(clusterNetwork, {"negative": "affinity"}) # čvorovi iz susednih klastera povezani su negativnim linkom
    print("Kreirana je mreža klastera.")
    return clusterNetwork

```

## Kreiranje random mreža:

Metoda koja služi za generisanje random klasterabilnih i neklasterabilnih mreža, kao parametre prima broj čvorova grafa, 1 ili 0 u zavisnosti od toga da li želimo klasterabilnu ili neklasterabilnu mrežu i parametar p koji predstavlja verovatnoću.

Metoda je implementirana tako da kreira prazan graf u koji zatim doda sve čvorove. Prvo prolazimo kroz sve čvorove i dodajemo im oznaku klastera kom pripadaju.

Ukoliko je potrebno kreirati klasterabilnu mrežu, prvo prolazimo kroz čvorove i proveravamo da li se nalaze u istom klasteru. Čvorovi koji se nalaze u istom klasteru moraju biti povezani pozitivnim linkom. Proveravamo da li čvor ima manje od 50 suseda (zbog velikog broja linkova koji se formira bez ovog ograničenja) i ukoliko ima, dodajemo pozitivan link. Ukoliko se čvorovi ne nalaze u istom klasteru, povezujemo ih negativnim linkom ako je verovatnoća veća od slučajne promenljive.

```
def randomGraph1(numNodes, clusterable, numClusters, p):
    graph = nx.Graph()
    graph.add_nodes_from(range(numNodes))

    if clusterable == 1:
        for n in range(numNodes):
            klaster = rand.randint(0, numClusters)
            graph.nodes[n]['c'] = klaster #dodela oznake klastera cvorovima

        for i in graph.nodes:
            for j in graph.nodes:
                if graph.nodes[i]['c'] == graph.nodes[j]['c']: #ukoliko se cvorovi nalaze u istom klasteru povezani su pozitivnom granom
                    lista = list(nx.neighbors(graph, i))
                    if len(lista) < 50:
                        affinity = "positive"
                        graph.add_edge(i, j, affinity = 'positive')
                else:
                    x = rand.randint(0, 1)
                    if p > x:
                        lista = list(nx.neighbors(graph, i))
                        if len(lista) < 5:
                            affinity = "negative"
                            graph.add_edge(i, j, affinity = 'negative')
```

Ukoliko je potrebno kreirati neklasterabilnu mrežu, čvorovima koji pripadaju istom klasteru dodeljujemo pozitivan link ako je verovatnoća veća od slučajne promenljive, u suprotnom, dodeljujemo negativan link. Ako čvorovi ne pripadaju istom klasteru, povezujemo ih negativnim linkom.

```

else:
    for n in range(numNodes):
        klaster = rand.randint(0, numClusters) #oznaka klastera
        graph.nodes[n]['c'] = klaster #dodela oznake covrovima

    for i in graph.nodes:
        for j in graph.nodes:
            x = rand.randint(0, 1)

            if graph.nodes[i]['c'] == graph.nodes[j]['c']:
                if p > x:
                    graph.add_edge(i, j, affinity = 'positive')
                else:
                    graph.add_edge(i, j, affinity = 'negative')
            else:
                lista = list(nx.neighbors(graph, i))
                if len(lista) < 5:
                    affinity = "negative"
                    graph.add_edge(i, j, affinity = 'negative')

print("Graf je ucitan.")
br = graph.number_of_nodes()
grana = graph.number_of_edges()
print("Broj cvorova u grafu: ",br)
print("Broj grana u grafu: ",grana)
return graph

```

## Primer jednog izvršavanja programa nad random grafovima:

### 1. Neklasterabilna mreža ( $p = 0.5$ )

```
Izaberite graf koji zelite da ucitate:
1 -> Epinions
2 -> Slashdot
3 -> Wiki
4 -> Klasterabilan(test)
5 -> Neklasterabilan(test)
6 -> Random generisan graf
7 -> Bitcoinalfa

Unesite jednu od ponuđenih mogućnosti: 6

-----Analiza random nastale mreze-----

Graf je ucitan.
Broj cvorova u grafu: 10000
Broj grana u grafu: 74595
Graf sadrzi 1074 klastera.
Napravljeno je 1074 grafova klastera.
Ovaj graf nije klasterabilan.
Postoji 24512 grana koje narušavaju klasterabilnost.
Da li zelite da se prikazu? (y/n)n
Broj povezanih komponenti: 1
Postoji 87 koalicija i 987 antikoalicija.
Da li zelite da se prikazu koalicije i antikoalicije? (y/n)n
Da li zelite analizu koalicija i antikoalicija? (y/n)n
```

### 2. Klasterabilna mreža ( $p = 0.5$ )

```
Izaberite graf koji zelite da ucitate:
1 -> Epinions
2 -> Slashdot
3 -> Wiki
4 -> Klasterabilan(test)
5 -> Neklasterabilan(test)
6 -> Random generisan graf
7 -> Bitcoinalfa

Unesite jednu od ponuđenih mogućnosti: 6

-----Analiza random nastale mreze-----

Graf je ucitan.
Broj cvorova u grafu: 10000
Broj grana u grafu: 74721
Graf sadrzi 1001 klastera.
Napravljeno je 1001 grafova klastera.
Ovaj graf je klasterabilan.
Broj povezanih komponenti: 1
Postoji 1001 koalicija i 0 antikoalicija.
Da li zelite da se prikazu koalicije i antikoalicije? (y/n)n
Da li zelite analizu koalicija i antikoalicija? (y/n)n
```



## Testiranje i primena na realnim mrežama:

### Analiza Bitcoin Alpha mreže:

Rezltati izvršavanja programa nad mrežom Bitcoin Alpha (soc-sign-bitcoinalpha.csv) koja predstavlja mrežu ljudi koji trguju bitkoinom na platformi Bitcoin Alpha.

```
Izaberite graf koji zelite da ucite:
```

- 1 -> Epinions
- 2 -> Slashdot
- 3 -> Wiki
- 4 -> Klasterabilan(test)
- 5 -> Neklasterabilan(test)
- 6 -> Random generisan graf
- 7 -> Bitcoinalfa

```
Unesite jednu od ponuđenih mogućnosti: 7
```

```
-----Analiza mreze Bitcoin-----
```

```
Broj cvorova u grafu: 3783
```

```
Broj grana u grafu: 24186
```

```
Graf sadrzi 122 klastera.
```

```
Napravljeno je 122 grafova klastera.
```

```
Ovaj graf nije klasterabilan.
```

```
Postoji 1213 grana koje narusavaju klasterabilnost.
```

```
Da li zelite da se prikazu? (y/n)n
```

```
Broj povezanih komponenti: 5
```

```
Postoji 121 koalicija i 1 antikoalicija.
```

```
Da li zelite da se prikazu koalicije i antikoalicije? (y/n)n
```

```
Kreiranje mreze klastera...
```

```
Kreirana je mreza klastera.
```

```
Analiza mreze klastera:
```

```
Broj cvorova: 122
```

```
Broj grana: 117
```

```
Gustina mreze klastera: 0.015851510635415254
```

```
Broj povezanih komponenti: 5
```

```
Nije moguće izracunati dijametar zbog postojanja vise komponenti povezanosti.
```

```
Prosecan stepen: 1.9180327868852458
```

```
Da li zelite analizu koalicija i antikoalicija? (y/n)y
```

```
Antikoalicije su kohezivnije od koalicija.
```

```
Antikoalicije imaju veci prosečni dijametar.
```

```
Antikoalicije su redje od koalicija.
```

Da li zelite da se prikazu koalicije i antikoalicije? (y/n)y

Koalicije i antikoalicije:

Koalicije:

['7469']

['7422']

['7365']

['7567']

['7546']

['7458']

['7350']

['7332']

['7548']

['7345']

['7495']

['7476']

['7370']

['7456']

['7359']

['7534']

['7477']

['7490']

['7384']

['7368']

['7474']

['7408']

['7459']

['7497']

['7475']

['7399']

['1870', '3271']

['7496']

['5837']

['7329']

['7376']

['7489']

['5029']

['7363']

['7467']

['7364']

['7339']

['7453']

['7485']

['7362']

['7346']

['7354']

```
('1243', '7508')
('7389', '3097')
('1020', '7583')
('193', '4057')
('193', '7388')
('6369', '2396')
('124', '3079')
('139', '2257')
('1830', '7554')
('233', '3428')
('7579', '933')
('403', '1159')
('385', '7570')
('7570', '425')
('393', '7410')
('1581', '7420')
('7413', '225')
('7394', '7520')
('7520', '791')
('7525', '335')
('7436', '3270')
('748', '564')
('3079', '1811')
('518', '1859')
('591', '7528')
('541', '7519')
('7528', '1159')
('973', '1159')
('1036', '7515')
('1522', '7506')
('7409', '791')
('791', '7414')
('791', '7396')
('791', '7395')
('1024', '7575')
('7375', '892')
('7416', '1778')
('619', '7561')
('7569', '1935')
('7569', '1930')
('496', '7558')
('7442', '2866')
('1153', '7411')
('7390', '2391')
```

Prikazane su grane koje narušavaju klasterabilnost grafa ( 1213 ).

## Analiza Wiki mreže:

Rezultati izvršavanja programa nad mrežom Wiki (wiki-RfA.txt) u kojoj čvorovi predstavljaju članove Wikipedia mreže, a linkovi glasove.

```
Izaberite graf koji zelite da ucitete:
1 -> Epinions
2 -> Slashdot
3 -> Wiki
4 -> Klasterabilan(test)
5 -> Neklasterabilan(test)
6 -> Random generisan graf
7 -> Bitcoinalfa

Unesite jednu od ponuđenih mogućnosti: 3

-----Analiza mreze Wikipedia-----

Broj cvorova u grafu: 11380
Broj grana u grafu: 189003
Graf sadrzi 2672 klastera.
Napravljeno je 2672 grafova klastera.
Ovaj graf nije klasterabilan.
Postoji 51266 grana koje narušavaju klasterabilnost.
Da li zelite da se prikazu? (y/n)n
Broj povezanih komponenti: 1
Postoji 2671 koalicija i 1 antikoalicija.
Da li zelite da se prikazu koalicije i antikoalicije? (y/n)n
Kreiranje mreze klastera...
Kreirana je mreza klastera.
Analiza mreze klastera:
Broj cvorova: 2672
Broj grana: 2808
Gustina mreze klastera: 0.0007868949484034552
Broj povezanih komponenti: 1
Dijametar: 4
Prosecan stepen: 2.1017964071856285
Da li zelite analizu koalicija i antikoalicija? (y/n)y
Antikoalicije su kohezivnije od koalicija.
Antikoalicije imaju veci prosečni dijametar.
Koalicije su redje od antikoalicija.
```

Antikoalicije:  
 ['Kingboyk', 'logan', 'GiantSnowman', 'I', 'Alex S', 'Daveydeeb', 'Kw', 'Ambu.Saxena', 'Secret', 'Ktr101', 'Thunderbrand', 'Nothing444', 'Ecoleetage', 'BQZip01', 'Zsinj', 'Bbb23', 'Redrose64', 'Pschemp', 'Cyde', 'Crum375', 'Alex Bakharev', 'Ynhockey', 'Kafziel', 'MatthewJND', 'AKGhetto', 'Ian13', 'Babajobu', 'Jeronimo', 'Deeputjoseph', 'Soumyasch', 'Nevi', 'Waldalstouring', 'Eyeserene', 'Until it sleeps', 'J Di', 'NawlinWiki', 'Phantomsteve', 'Glores', 'Pedro', 'Ezeamon', 'Magnus animus', 'Malik Shabazz', 'Graft', 'Number 57', 'Paolo Liberatore', 'HD Mitchell', 'Derek Ross', 'dave souza', 'Xivexgoem', 'Ragesoss', 'Gwernol', 'Chacor', 'Skater', 'Altruism', 'Drilnoth', 'Commander Keane', 'Pastordavid', 'LordAmeth', 'J.del anoy', 'Gactusman', 'Blightpeace', 'Nishkide4', 'Sam Blacketer', 'Errabse', 'Ice cold Beer', 'T-dot', 'Oxbus', 'Schutz', 'Trempley', 'Killarvogel', 'Cp013', 'The utahraptor', 'Saravask', 'Bkthow', 'Stove02', 'Sam Spade', 'Pasto', 'Cholmes76', 'Tonythetiger', 'Kcordina', 'Tollyaddict', 'TravistX', 'Canadian Bacon', 'Renato2', 'Jza84', 'Miketych', 'HereIHelp', 'Heligoland', 'Graeme Bartlett', 'Tillwe', 'shoofdeath', 'RevRagnarok', 'Addihockey10', 'Dsmgold', 'William', 'Accdude92', 'Bobo192', 'Qst', 'Epb123', 'Appleboy', 'Alexandromitri', 'Jogers', 'Bjelleklang', 'RyanLupin', 'Hex', 'Aarons', 'Darkfalls', 'Tregoweth', 'Stuffofinterest', 'Stephen B Streater', 'Jesse Viviano', 'Synergetichagot', 'Bellhalla', 'Idleguy', 'Mysid', 'Todor0zhinov', 'Pepsidrinka', 'Earthbendingmaster', 'Khoikhoi', 'DaGizza', 'Prashanthns', 'Happyne22', 'Travellingcari', 'DragonflySixtyseven', 'ScottDavis', 'Mattythewhit e', 'Meelar', 'IlovePlankton', 'Kudpung', 'Joshbuddy', 'Wily0', 'Matthewfenton', 'RayAYang', 'J04n', 'Scm83x', 'Terence Ong', 'Jasper Deng', 'Anonymous editor', 'Carrite', 'Cesar8', 'M r. IP', 'EliZ81', 'Wonderfool', 'Martijn Hoekstra', 'Anthony.bradbury', 'Ezeu', 'SarekOfVulcan', 'Andrew Levine', 'Mr Senseless', 'Tedernst', 'The G-Unit-Boss', 'How do you turn this o n', 'Roux', 'Firefoxman', 'Georgewilliamherbert', 'Ali'i', 'Encyclopedist', 'Zachary crimsonwolf', 'Jake Wartenberg', 'Ceranthon', 'Xenocidic', 'Tiptoeity', 'Phaedriel', 'Shapiro10', 'ShaneKing', 'Eustress', 'Y0u', 'Magioladitis', 'Dev920', 'Haham hanuka', 'JD UK', 'Dustinagic', 'Laralove', 'QuiteUnusual', 'Consumed Crustacean', 'Rogerthat', 'Freakofnuture', 'Wisdom89', 'J Milburn', 'Richardshusr', 'Kimchi.sg', 'Ryulong', 'Tangotango', 'Xchrisblackx', 'Lord Voldeamort', 'The Blade of the Northern Lights', 'Biekk0', 'RedirectCleanupBot', 'Bastique m89', 'James886', 'Jay147', 'HighwayCello', 'Txman307', 'Andrusc', 'UninvitedCompany', 'Gwen Gale', 'Tehkewll', 'Yacht', 'JPG-GR', 'Fribbler', 'Robchurch', 'Nandesuka', 'Richwales', 'MON GO', 'Information Center', 'Smaddy', 'Wikipediainfo2221', 'Rgoodemote', 'Rettetast', 'coren', 'Tawker', 'Rmfstar', 'Paul Cyr', 'Staxringold', 'Blueboy96', 'Ginkgo100', 'M2McBride', 'Choess', 'Physic210', 'Flotlight', 'Acetic Acid', 'Jbmurray', 'Ohanaunited', 'LAX', 'Geni', 'General Eisenhower', 'Humblefool', 'Bookofjude', 'Nthepp', 'Henrygb', 'Ramallite', 'Ligulem', 'HaeMo', 'Flowerparty', 'Siroxo', 'Kbb3rd', 'Kuru', 'The Bushranger', 'Patrick', 'Visorsstuff', 'Aitias', 'Brian0918', 'Zedla', 'Evad0', 'AGK', 'Die22', 'Pyleslong', 'Bhadani', 'Reed y Boy', 'Abd', 'Haurreen', 'Dipankar001', 'Chzz', 'Weirdy', 'Brownairedgill', 'Chamal N', 'Stephan Schulz', 'Editorofthewiki', 'Andrew e', 'Bduke', 'Itszippy', 'Karmafist', 'Kilo-Lima', 'Physch162', 'MGA73', 'Lquilter', 'Maeblis', 'RockMagnetist', 'Fetchcomes', 'JRM', 'JDoorjam', 'Bradjamesbrown', 'L Aquatique', 'Cenarium', 'TommyBoy', 'Wtmitchell', 'The Transhumanist', 'ZooPro', 'King of Hearts', 'Katerenka', 'Halibutt', 'SebWite', 'computerjoe', 'DrKiernan', 'Ocaasi', 'MADAP', 'AngusmcLellan', 'Fa', 'Pagrashak', 'connormah', 'Soxred93', 'M0RS', 'Scetoax', 'KI', 'Karanacs', 'Rama's Arrow', 'Turnstep', 'Raven4x4x', 'Srikeit', 'Catanomphisa', 'LeonardBloom', 'Ngh', 'Suntag', 'Garzo', 'Shreshth01', 'Theopolisme', 'Sir jam es paul', 'Tango', 'Alansohn', 'David Levy', 'Gien', 'Guinnog', 'Samuel Blanning', 'Navou', 'TKD', 'Dangherous', 'Mached', 'Nat', 'Bottesini', 'Gordonrox24', 'Ymblanter', 'Majorly', 'Deckiller', 'Pmanderson', 'Headbomb', 'Jason Quinn', 'Aldux', 'Guest9999', 'Extraordinary Machine', 'Dewet', 'Yandman', 'Sarge Baldy', 'Vary', 'Pkg', 'Wikifanatic', 'LuK3', 'Krimpet', 'Imperatorexercitus', 'Celestianpower', 'Blanchardb', 'Backslash Forwardslash', 'Kwsn', 'Kjkolb', 'Ironholds', 'DGG', 'Cookiecaper', 'Hu12', 'FT2', 'Werdna', 'Radiokirk', 'Fetofs', 'Avr uch', 'Ikiroid', 'Zappa.jake', 'Ikinias', 'The c of E', 'Groove0og', 'Anonymous Dissident', 'Avar Arnfjörð Bjarnason', 'ChrisGriswold', 'Paul Erik', 'Joelr31', 'TomStar81', 'Seicer', 'Pulltoopen', 'Bornhj', 'Guogu012', 'Pax', 'Stifle', 'Vejvancicky', 'Cireland', 'Addshore', 'Humus sapiens', 'Kotra', 'Dfrg.msc', 'Kaisershatner', 'Basket of Puppies', 'Markovich292', 'Musical Linguist', 'Vantucky', 'CBDunkerson', 'Gaius Cornelius', 'Mr. Lefty', '(aeropagitica)', 'Hut 8.5', 'Mabdul', 'The Thing That Should Not Be', 'Hadal', 'David Fuchs', 'Elonka', 'Misza13', 'Samsara', 'Greeves', 'RGTraynor', 'Mets501', 'William Pietri', 'Hall Monitor', 'Joe Beaudoin Jr.', 'Slon02', 'Peruvianllama', 'Nativeforeigner', 'Kingoomielii', 'Malleus Fat uarum', 'Kylu', 'Hennessey', 'Patrick', 'Trialsanderrors', 'Sarah Ewart', 'Dreamafter', 'Poeloo', 'Philosopher', 'Fw', 'Canadian Paul', 'The Halo', 'Pfortuny', 'Triona', 'Lar', 'TomasBa t', 'J.smith', 'Yao Ziyuan', 'Selket', 'Janesontai', 'Adam213', '23skidoo', 'KFP', 'No guru', 'Future Perfect at Sunrise', 'Matthewedwards', 'Sean Black', 'MCDetroit', 'Kelapstick', 'Wenli', 'Bangerrrior', 'Guamco', 'Breaycker', 'Hijuna Brass', 'Everyking', 'Freddie', 'Lord Roem', 'Mboverload', 'Willing1979', 'Punster21', 'Jedifg', 'Nuclearwarfare', 'Marianococo wski', 'Lightdarkness', 'Stwalkerster', 'Black Falcon', 'Bwilkins', 'Camillo', 'Bjweeks', 'Who', 'Ronnotel', 'Cleared as filled', 'Alphax', 'Shimeru', 'Maddin', 'Makemi', 'Re aper Eternal', 'Trebore Rowntree', 'ZooFari', 'Uwrf', 'Canadiancaesar', 'Teckwiz', 'Wikidudeman', 'Stevetigo', 'Staffwaterboy', 'Christopher Mahan', 'Sverdrup', 'Gryffindor', 'Legoktm', 'Aranda56', 'Lexor', 'J-stan', 'Smashville', 'Irishguy', 'Rune.welsh', 'Bubba hotep', 'Riddle', 'Krosenzweig', 'Dark Dragon Flame', 'Howabout1', 'Dank55', 'SNCandlish', 'Nikki311', 'Simetrical', 'Vibhijain', 'Dreadstar', 'Redvers', 'Daniel Olsen', 'TheCatalyst31', 'Texture', 'Pjacobi', 'Arvindn', 'Aq01rius', 'Scottywyong', 'CTShynken', 'Horologium', 'Clarkk', 'A Link to the Past', 'MASCAR Fan24', 'Moni3', 'Oscarthecat', 'Minimac', 'Savidan', 'Master Jay', 'Martial Law', 'Velyos', 'Buckshot06', 'Ottava Rima', 'TimVickers', 'Phils', 'Eluchil 404', 'Wikieditor222', 'Antandrus', 'Flyingtoaster', 'Wikihermit', 'Madman', 'Porchcrop', 'Evilclown93', 'Mike Selinker', 'Explicit', 'Sasquatch', 'Mikel', 'Jay', 'AVArkots', 'R.koot', 'Loom01', 'tjstfr', 'Natalie Erin', 'CJLL Wright', 'Betacommand', 'R', 'Rklawton', 'Shirik', 'Joshua2', 'Shadow', 'Chris 73', 'Hersfold', 'A new name 2008', 'Protokn', 'Winhunter', 'Adambiswanger1', 'Robert McLenon', 'Iustiger seth', 'Tassedetthe', 'Cb162', 'Hader', 'Alkivar', 'The JPS', 'Marcus Quertus', 'Floquenbeam', 'Getcrunk', 'Euryalus', 'Lradrama', 'Ademo n', 'Nakonkantari', 'Rory096', 'Hurricanehink', 'TwinsMetsFan', 'Proto', 'CRGreathouse', 'Mark Richards', 'SLSB', 'Soman', 'Persian Poet Gal', 'Sirfozzie', 'Danielfolsom', 'Katefan0', 'S147', 'Mehrschad123', 'Herostratus', 'Airplaneman', 'SoothingR', 'Kurt Shaped Box', 'White Shadows', 'Sigrandson', 'Obli', 'Andreasegde', 'Snoutwood', 'Keenan Pepper', 'Gorillawarfare', 'IronDuke', 'Dennis Brown', 'Tone', 'Lawrence Cohen', 'Kizor', 'Tanthalas39', 'Milk's Favorite cookie', 'Vicenarian', 'Joturner', 'Dalbury', 'Gator1', 'Dertexer', 'Cailli', 'Nikkima

## Koalicije:

['Highfields']  
 ['Adam Pirolo']  
 ['KuatofKDY']  
 ['Benmay']  
 ['43?enter']  
 ['Steal']  
 ['Azythia Goes On']  
 ['Gvor1']  
 ['Multixer']  
 ['Central Data Bank']  
 ['B3virq3b']  
 ['Ant682']  
 ['Hanuman Das']  
 ['Goplat']  
 ['Chiacomo']  
 ['Luckytoilet']  
 ['Kandal the Vandal']  
 ['Arch NME']  
 ['KenDenier']  
 ['Schzmo3']  
 ['Eridani']  
 ['Alterego']  
 ['Hakozen']  
 ['Oreos']  
 ['norge0209']  
 ['ElectricEye']  
 ['Admiral Roo']  
 ['Poulsen']  
 ['LittleBenW']  
 ['Skinnyweed']  
 ['Danielaustinhall12']  
 ['Darth Mike']  
 ['068152']  
 ['Gol']  
 ['Wellmann']  
 ['nabilmmezher']  
 ['Terrybader']  
 ['Numsk11']  
 ['Froth']  
 ['Geoffrey Mitchell']  
 ['Frutti di Mare']  
 ['maxviwe']  
 ['Mrmoocow']  
 ['2D']  
 ['YOSF0113']

```
('A3RO', 'Equazcion')
('PNiddy', 'Lucid')
('PNiddy', 'Kkrouni')
('Slicky', 'Mikkerpikker')
('Kamope', 'Scharks')
('Kamope', 'Oliver202')
('Parker007', 'Skittle')
('Parker007', 'Wooty')
('NYC JD', 'Scharks')
('Jhfireboy', 'LeCour')
('Pcassidy', 'Davetunney')
('Dragomiloff', 'Scharks')
('ONUnicorn', 'Oliver202')
('Widr', 'Vacation9')
('TBrandley', 'Darkness Shines')
('Go Phightins!', 'Darkness Shines')
('Go Phightins!', 'Vacation9')
('Cncmaster', 'Vacation9')
('Intothatdarkness', 'Hghyux')
('Mediran', 'Vacation9')
('Tazerdadog', 'Vacation9')
('Vacation9', 'JayJay')
('Vacation9', 'LlamaAl')
('Vacation9', 'Riley Huntley')
('Vacation9', 'Aircorn')
('Vacation9', 'Greengreengreenred')
('Vacation9', 'Retrolord')
('Vacation9', 'John F. Lewis')
('A Train', 'Kizzle')
('Hghyux', 'Tomtomn00')
('ScotchMB', 'Daniel 123')
('Darkness Shines', 'Huon')
('Scharks', 'SciAndTech')
('GT', 'The Republican')
('CDThieme', 'YHoshua')
('Huaiwei', 'Instantnood')
('The Republican', 'Vulcanstar6')
('Johnny Au', 'Runewiki777')
('Bkwillwm', 'LordViD')
('Kinneyboy90', 'Daniel 123')
('Purplebackpack89', 'Josh Parris')
('New England', 'Perspicacite')
('WikiCats', 'Argentino')
('Zink Dawg', 'Pickbothmanlol')
Prikazane su grane koje narusavaju klasterabilnost grafa ( 51266 ).
```

## Analiza Slashdot mreže:

Izaberite graf koji zelite da ucitate:

- 1 -> Epinions
- 2 -> Slashdot
- 3 -> Wiki
- 4 -> Klasterabilan(test)
- 5 -> Neklasterabilan(test)
- 6 -> Random generisan graf
- 7 -> Bitcoinalfa

Unesite jednu od ponuđenih mogućnosti: 2

-----Analiza mreze Slashdot-----

Broj cvorova u grafu: 77350

Broj grana u grafu: 516575

Graf sadrzi 7286 klastera.

Napravljeno je 7286 grafova klastera.

Ovaj graf nije klasterabilan.

Postoji 103269 grana koje narušavaju klasterabilnost.

Da li zelite da se prikazu? (y/n)n

Broj povezanih komponenti: 1

Postoji 7285 koalicija i 1 antikoalicija.

Da li zelite da se prikazu koalicije i antikoalicije? (y/n)n

Kreiranje mreze klastera...

Kreirana je mreza klastera.

Analiza mreze klastera:

Broj cvorova: 7286

Broj grana: 7376

Gustina mreze klastera: 0.00027792792224197704

Broj povezanih komponenti: 1

Dijametar: 4

Prosecan stepen: 2.0247049135328026