СОЦИЈАЛНЕ МРЕЖЕ

НИКОЛА ФИМИЋ 2019

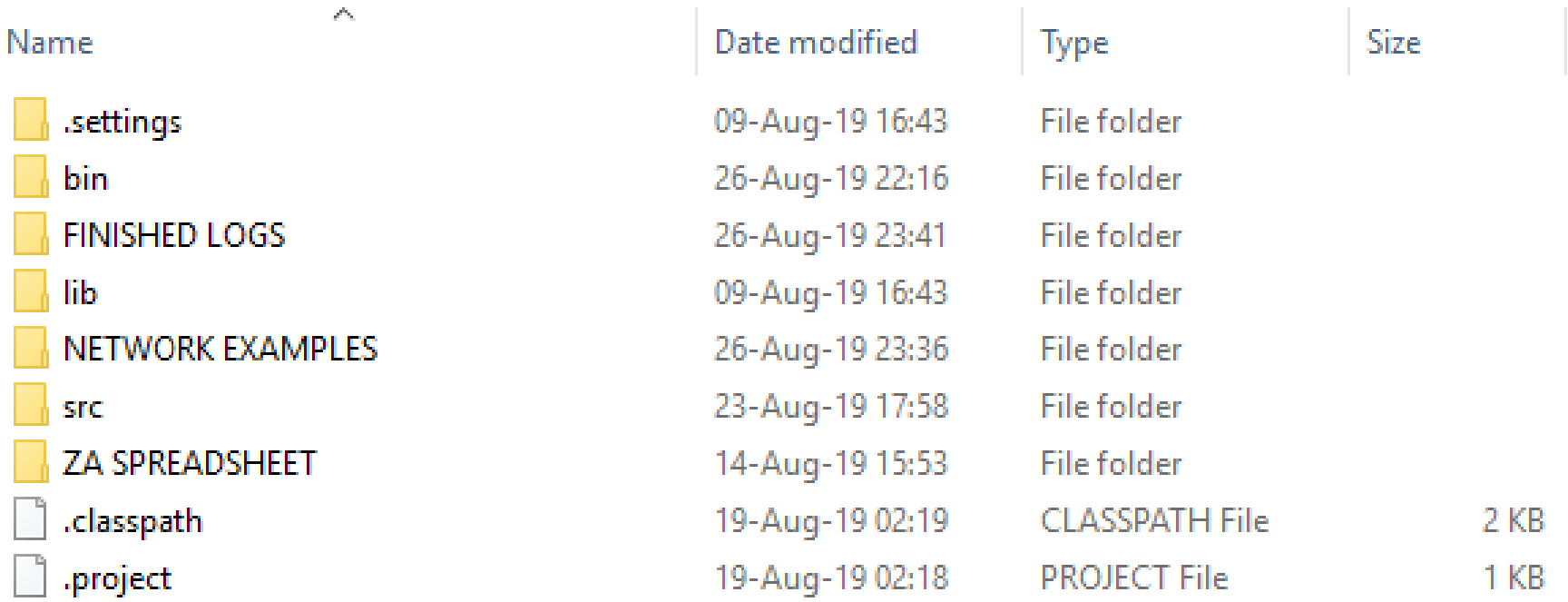
ОРГАНИЗАЦИЈА ФАЈЛОВА

FOLDER NETWORK EXAMPLES – ФОЛДЕР СА ПРИМЕРИМА РАЗНИХ МРЕЖА

FOLDER FINISHED LOGS – ФОЛДЕР СА ОБРАЂЕНИМ МРЕЖАМА

FOLDER ZA SPREADSHEET – ФОЛДЕР У КОЈЕМ СЕ НАЛАЗЕ ФАЈЛОВИ ПОТРЕБНИ ЗА ИМПОРТОВАЊЕ МРЕЖЕ У GEPHI РАДИ ВУЗУАЛИЗАЦИЈЕ

ОРГАНИЗАЦИЈА ФАЈЛОВА



Name	Date modified	Type	Size
.settings	09-Aug-19 16:43	File folder	
bin	26-Aug-19 22:16	File folder	
FINISHED LOGS	26-Aug-19 23:41	File folder	
lib	09-Aug-19 16:43	File folder	
NETWORK EXAMPLES	26-Aug-19 23:36	File folder	
src	23-Aug-19 17:58	File folder	
ZA SPREADSHEET	14-Aug-19 15:53	File folder	
.classpath	19-Aug-19 02:19	CLASSPATH File	2 KB
.project	19-Aug-19 02:18	PROJECT File	1 KB

VERTICE

- КЛАСА КОЈА ПРЕДСТАВЉА ЈЕДНУ ТАЧКУ У МРЕЖИ
- ГЕНЕРИЧНА КЛАСА
- GETTERI I SETTERI
- TOSTRING METOD
- HASHCODE
- EQUALS

```java
public class Vertice<T> {

    T info;

    public Vertice() {}

    public Vertice(T info) {
        this.info = info;
    }

    public T getInfo() {
        return info;
    }

    public void setInfo(T vertice) {
        this.info = vertice;
    }

    @Override
    public String toString() {
        return info + "";
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + ((info == null) ? 0 : info.hashCode());
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Vertice other = (Vertice) obj;
        if (info == null) {
            if (other.info != null)
                return false;
        } else if (!info.equals(other.info))
            return false;
        return true;
    }
```

EDGE

- КЛАСА КОЈА ПРЕДСТАВЉА ЈЕДНУ ВЕЗУ У МРЕЖИ
- BOOLEAN POSTIVE RELATION
- GETTERI I SETTERI
- TOSTRING METODI

```java
public class Edge {

    boolean positiveRelation;
    Vertice sourceVertice;
    Vertice targetVertice;

    public Edge(boolean positiveRelation, Vertice sourceVertice, Vertice targetVertice) {
        this.positiveRelation = positiveRelation;
        this.sourceVertice = sourceVertice;
        this.targetVertice = targetVertice;
    }

    public boolean getPositiveRelation() {
        return positiveRelation;
    }

    public Vertice getSourceVertice() {
        return sourceVertice;
    }

    public void setSourceVertice(Vertice sourceVertice) {
        this.sourceVertice = sourceVertice;
    }

    public Vertice getTargetVertice() {
        return targetVertice;
    }

    public void setTargetVertice(Vertice targetVertice) {
        this.targetVertice = targetVertice;
    }

    public void setPositiveRelation(boolean positiveRelation) {
        this.positiveRelation = positiveRelation;
    }

    @Override
    public String toString() {
        String relation = positiveRelation == true ? "+" : "-";
        //return relation + "[" + sourceVertice + "," + targetVertice + "]";
        return relation;
    }

    public String toStringWithVertices() {
        String relation = positiveRelation == true ? "+" : "-";
        return relation + "[" + sourceVertice + "," + targetVertice + "]";
    }
```

RANDOM NETWORK GENERATOR

```
public class RandomNetworkGenerator {
    int numberOfVertices;
    int numberOfNamesForVertices = 1100;

    public RandomNetworkGenerator(int numberOfVertices) throws Exception {
        if(this.numberOfNamesForVertices >= numberOfVertices) {
            BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("NETWORK EXAMPLES/RANDOM mreza.txt")));
            BufferedWriter writerForNodes = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("ZA SPREADSHEET/nodes.txt")));
            BufferedWriter writerForEdges1 = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("ZA SPREADSHEET/edge1.txt")));
            BufferedWriter writerForEdges2 = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("ZA SPREADSHEET/edge2.txt")));
            BufferedWriter writerForRelation = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("ZA SPREADSHEET/relation.txt")));


            Random random = new Random();
            this.numberOfVertices = numberOfVertices;
            Set<Integer> nodesSet = new HashSet<>();

            while(nodesSet.size() < numberOfVertices) {
                int number1 = random.nextInt(numberOfNamesForVertices);
                int number2 = random.nextInt(numberOfNamesForVertices);

                if(number1 == number2) {
                    while(number1 == number2) {
                        number2 = random.nextInt(numberOfNamesForVertices);
                    }
                }

                nodesSet.add(number1);
                nodesSet.add(number2);

                int relationInt = random.nextInt(10);
                boolean relationBoolean = relationInt < 8 ? true : false;
                String relationString = relationBoolean == true ? "+" : "-";

                writer.write(number1 + " " + number2 + " " + relationString + System.lineSeparator());
                writerForEdges1.write(number1 + System.lineSeparator());
                writerForEdges2.write(number2 + System.lineSeparator());

                if(relationBoolean == true) {
                    writerForRelation.write("1" + System.lineSeparator());
                }
                else {
                    writerForRelation.write("0.1" + System.lineSeparator());
                }
```

ΚΛΑΣΑ CLUSTER FINDER

```
public class ClusterFinder<V, E> {
    Set<E> edgesToDeleteIfNotClusterable;
    Collection<Set<V>> NOTCoalitionClusterCollection;

    public Set<E> getEdgesToDeleteIfNotClusterable() {
        return edgesToDeleteIfNotClusterable;
    }

    public Collection<Set<V>> getNOTCoalitionClusterCollection() {
        return NOTCoalitionClusterCollection;
    }
```

МЕТОД КОЈИ ПРОНАЛАЗИ КЛАСТЕРЕ

```
public Set<Set<V>> findClusters(UndirectedSparseGraph<V, E> graph, Transformer<E, Boolean> transformer) {

    // REMOVE EDGES WITH NEGATIVE RELATION
    Collection<E> edges = graph.getEdges();
    Iterator<E> edgeIterator = edges.iterator();
    List<E> toRemove = new ArrayList<>();

    while (edgeIterator.hasNext()) {
        E edge = edgeIterator.next();
        if (transformer.transform(edge) == false) {
            toRemove.add(edge);
        }

    }

    for (E edge : toRemove) {
        graph.removeEdge(edge);
    }
    ////////////////////////////////////////////////////////////////////////////////


    //IDENTIFY COMPONENTS USING DFS
    DFSComponents<V, E> dfsComponents = new DFSComponents<>(graph);
    Set<Set<V>> components = dfsComponents.identifyComponents();

    return components;
    ////////////////////////////////////////////////////////////////////////////////
}
```

МЕТОД КОЈИ ПРОВЕРАВА ДА ЛИ ЈЕ МРЕЖА КЛАСТЕРАБИЛНА

```
public boolean isClusterable(UndirectedSparseGraph<V, E> backupGraph, UndirectedSparseGraph<V, E> graph, Set<Set<V>> clusters, Transformer<E, Boolean> transformer) {
    boolean isClusterableBoolean = true;
    edgesToDeleteIfNotClusterable = new HashSet<>();
    NOTCoalitionClusterCollection = new HashSet<>();

    //ITERATE THROUGH EACH CLUSTER
    Iterator<Set<V>> clusterIterator = clusters.iterator();
    while (clusterIterator.hasNext()) {
        Set<V> cluster = clusterIterator.next();
        Iterator<V> verticesInClusterIterator = cluster.iterator();

        //ITERATE THROUGH VERTICES OF EACH INDIVUDUAL CLUSTER
        while (verticesInClusterIterator.hasNext()) {
            V checkingVertice = verticesInClusterIterator.next();
            Iterator<V> verticesInClusterIterator2 = cluster.iterator();
            //AGAIN ITERATE THROUGH VERTICES OF CURRENT CLUSTER AND CHECK IF THERE IS A NEGATIVE EDGE BETWEEN ANY TWO VERTICES IN CURRENT CLUSTER
            while (verticesInClusterIterator2.hasNext()) {
                V verticeToCheck = verticesInClusterIterator2.next();
                if (backupGraph.findEdge(checkingVertice, verticeToCheck) != null
                        && transformer.transform(backupGraph.findEdge(checkingVertice, verticeToCheck)) == false) {
                    edgesToDeleteIfNotClusterable.add(backupGraph.findEdge(checkingVertice, verticeToCheck));
                    NOTCoalitionClusterCollection.add(cluster);
                    isClusterableBoolean = false;
                }
            }

        }

    }
    return isClusterableBoolean;
}
```

МЕТОД КОЈИ ВРАЋА КЛАСТЕРЕ ИЗ МРЕЖА КАО НОВЕ ГРАФОВЕ

```
public Collection<UndirectedSparseGraph<V, E>> clustersAsGraphs(UndirectedSparseGraph<V, E> graph, Set<Set<V>> clusters) {
    Collection<UndirectedSparseGraph<V, E>> clustersAsGraphsCollection = new ArrayList<>();
    Iterator<Set<V>> clustersIterator = clusters.iterator();

    while (clustersIterator.hasNext()) {
        Set<V> cluster = clustersIterator.next();

        UndirectedSparseGraph<V, E> clusterGraph = clusterAsGraph(graph, cluster);
        clustersAsGraphsCollection.add(clusterGraph);
    }
    return clustersAsGraphsCollection;

}

private UndirectedSparseGraph<V, E> clusterAsGraph(UndirectedSparseGraph<V, E> graph, Set<V> cluster) {
    UndirectedSparseGraph<V, E> clusterGraph = new UndirectedSparseGraph<>();

    //ITERATE THROUGH VERTICES OF CLUSTER
    Iterator<V> singleClusterIterator = cluster.iterator();
    while (singleClusterIterator.hasNext()) {
        //FOR EACH VERTICE, FIND INCIDENT EDGES
        V vertice = singleClusterIterator.next();
        Collection<E> incidentEdges = graph.getIncidentEdges(vertice);
        Iterator<E> incidentEdgesIterator = incidentEdges.iterator();
        //FOR EACH INCIDENT EDGE, FIND VERTICES WHICH IT CONNECTS AND PUT IT IN THE NEW GRAPH
        while (incidentEdgesIterator.hasNext()) {
            E edge = incidentEdgesIterator.next();
            Collection<V> incidentVertices = graph.getIncidentVertices(edge);

            clusterGraph.addEdge(edge, incidentVertices);
        }

    }
    return clusterGraph;
}
```

- КЛАСА У КОЈОЈ ЈЕ ИМПЛЕМЕТИРАН ДФС АЛГОРИТАМ

```java
public class DFSComponents<V, E> {
    Set<V> visited;
    Set<Set<V>> components;
    UndirectedSparseGraph<V, E> graph;


    public DFSComponents(UndirectedSparseGraph<V, E> graph) {
        this.graph = graph;
    }

    public Set<Set<V>> identifyComponents() {
        visited = new HashSet<>();
        components = new HashSet<>();

        for (V vertice : graph.getVertices()) {
            if(!visited.contains(vertice)) {
                components.add(identifyComponent(vertice));
            }
        }
        return components;
    }

    private Set<V> identifyComponent(V vertice) {
        Set<V> component = new HashSet<>();
        component.add(vertice);
        visited.add(vertice);

        dfs(vertice, component);

        return component;



    }

    private void dfs(V current, Set<V> component) {
        for (V neighbour : graph.getNeighbors(current)) {
            if(!visited.contains(neighbour)) {
                component.add(neighbour);
                visited.add(neighbour);
                dfs(neighbour, component);
            }
        }

    }
```

ΚΛΑΣΑ MAIN GRAPH

```
public class MainGraph {
    public static final DecimalFormat decimalFormat = new DecimalFormat("####0.0");

    public static void main(String[] args) throws Exception {
        long startTime = System.nanoTime();
        String fileName = "bitcoin trust network CLUSTERABLE.txt";


        UndirectedSparseGraph<Vertice, Edge> graph = new UndirectedSparseGraph<>();
        UndirectedSparseGraph<Vertice, Edge> backupGraph = new UndirectedSparseGraph<>();
        BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(new FileOutputStream("FINISHED LOGS/" + fileName)));

        // GENERATING RANDOM NETWORK
        RandomNetworkGenerator randomNetworkGenerator = new RandomNetworkGenerator(1000);
        ////////////////////////////////////////////////////////////////////////////////


        // RELATION TRANSFORMER FOR EDGES
        Transformer<Edge, Boolean> edgeRelationTransformer = new Transformer<Edge, Boolean>() {
            public Boolean transform(Edge edge) {
                return edge.getPositiveRelation();
            }
        };
        ////////////////////////////////////////////////////////////////////////////////
```

МЕТОДИ ЗА ЧИТАЊЕ МРЕЖЕ ИЗ ФАЈЛА

```
public static void readBitcoinTrustNetwork(String fileName, UndirectedSparseGraph<Vertice, Edge> graph, UndirectedSparseGraph<Vertice, Edge> backupGraph) throws Exception {

public static void readRandomNetwork(String fileName, UndirectedSparseGraph<Vertice, Edge> graph, UndirectedSparseGraph<Vertice, Edge> backupGraph) throws Exception {

public static void readNetwork(String fileName, UndirectedSparseGraph<Vertice, Edge> graph, UndirectedSparseGraph<Vertice, Edge> backupGraph) throws Exception {

public static void readWikiRequstForAdminshipNetwork(String fileName, UndirectedSparseGraph<Vertice, Edge> graph, UndirectedSparseGraph<Vertice, Edge> backupGraph) throws Exception {

public static void readRedditNetwork(String fileName, UndirectedSparseGraph<Vertice, Edge> graph, UndirectedSparseGraph<Vertice, Edge> backupGraph) throws Exception {
```

ГЕНЕРИСАЊЕ СКУПА КЛАСТЕРА И ПРОВЕРАВАЊЕ ДА ЛИ ЈЕ МРЕЖА КЛАСТЕРАБИЛНА

```
// GENERATE SET OF CLUSTERS
ClusterFinder<Vertice, Edge> clusterFinder = new ClusterFinder<>();
Set<Set<Vertice>> clusters = clusterFinder.findClusters(graph, edgeRelationTransformer);
////////////////////////////////////////////////////////////////////////////////

writer.write("Number of NODES in network = " + backupGraph.getVertexCount() + System.lineSeparator());
writer.write("Number of EDGES in network = " + backupGraph.getEdgeCount() + System.lineSeparator());
writer.write(System.lineSeparator());

// CHECK IF THE NETWORK IS CLUSTERABLE
boolean isClusterable = clusterFinder.isClusterable(backupGraph, graph, clusters, edgeRelationTransformer);
String isClusterableString = isClusterable ? "IS" : "IS NOT";
writer.write("The network " + isClusterableString + " clusterable" + System.lineSeparator());
writer.write("-----------------------------------------------------------------------------------------
writer.write(System.lineSeparator());
System.out.println(isClusterable);
////////////////////////////////////////////////////////////////////////////////
```

ПРОНАЛАЖЕЊЕ ВЕЗА КОЈЕ БИ ТРЕБАЛО ИЗБРИСАТИ ДА БИ МРЕЖА ПОСТАЛА КЛАСТЕРАБИЛНА

```
// IF THE NETWORK IS NOT CLUSTERABLE, FIND THE EDGES THAT NEED TO BE DELETED IN ORDER TO BECOME CLUSTERABLE
if(!isClusterable) {
    Set<Edge> getEdgesToDeleteIfNotClusterable = clusterFinder.getEdgesToDeleteIfNotClusterable();

    writer.write(getEdgesToDeleteIfNotClusterable.size() + " EDGES NEED TO BE DELETED");
    writer.write(System.lineSeparator());

    double percentOfEdgesToDelete = ((double) getEdgesToDeleteIfNotClusterable.size() / (double) backupGraph.getEdgeCount()) * 100;
    writer.write(decimalFormat.format(percentOfEdgesToDelete) + "% OF ALL EDGES NEED TO BE DELETED");
    writer.write(System.lineSeparator());

    Iterator<Edge> getEdgesToDeleteIfNotClusterableIterator = getEdgesToDeleteIfNotClusterable.iterator();
    writer.write("IN ORDER FOR THE NETWORK TO BECOME CLUSTERABLE, THESE EDGES NEED TO BE DELETED = " + System.lineSeparator());

    writer.write(System.lineSeparator());

    while (getEdgesToDeleteIfNotClusterableIterator.hasNext()) {
        Edge edgeToDelete = getEdgesToDeleteIfNotClusterableIterator.next();
        writer.write(edgeToDelete.toStringWithVertices() + System.lineSeparator());
    }
    writer.write("-----------------------------------------------------------------------------------------------------------------------");
    writer.write(System.lineSeparator());
}
////////////////////////////////////////////////////////////////////////////////
```

ПРЕТВАРАЊЕ КЛАСТЕРА У НОВЕ САМОСТАЛНЕ МРЕЖЕ

```
// GENERATE GRAPHS FROM CLUSTERS
Collection<UndirectedSparseGraph<Vertice, Edge>> clustersAsGraphsCollection = clusterFinder.clustersAsGraphs(graph, clusters);
int i = 1;

for (UndirectedSparseGraph<Vertice, Edge> cluster : clustersAsGraphsCollection) {
    if(cluster.getVertexCount() < 2) {
        continue;
    }
    writer.write("CLUSTER " + i + "(" + cluster.getVertexCount() + " vertices) = " + cluster.toString() + System.lineSeparator());
    writer.write(System.lineSeparator());
    i++;
}
writer.write("----------------------------------------------------------------------------------------------------------------------------
writer.write(System.lineSeparator());
////////////////////////////////////////////////////////////////////////////
```

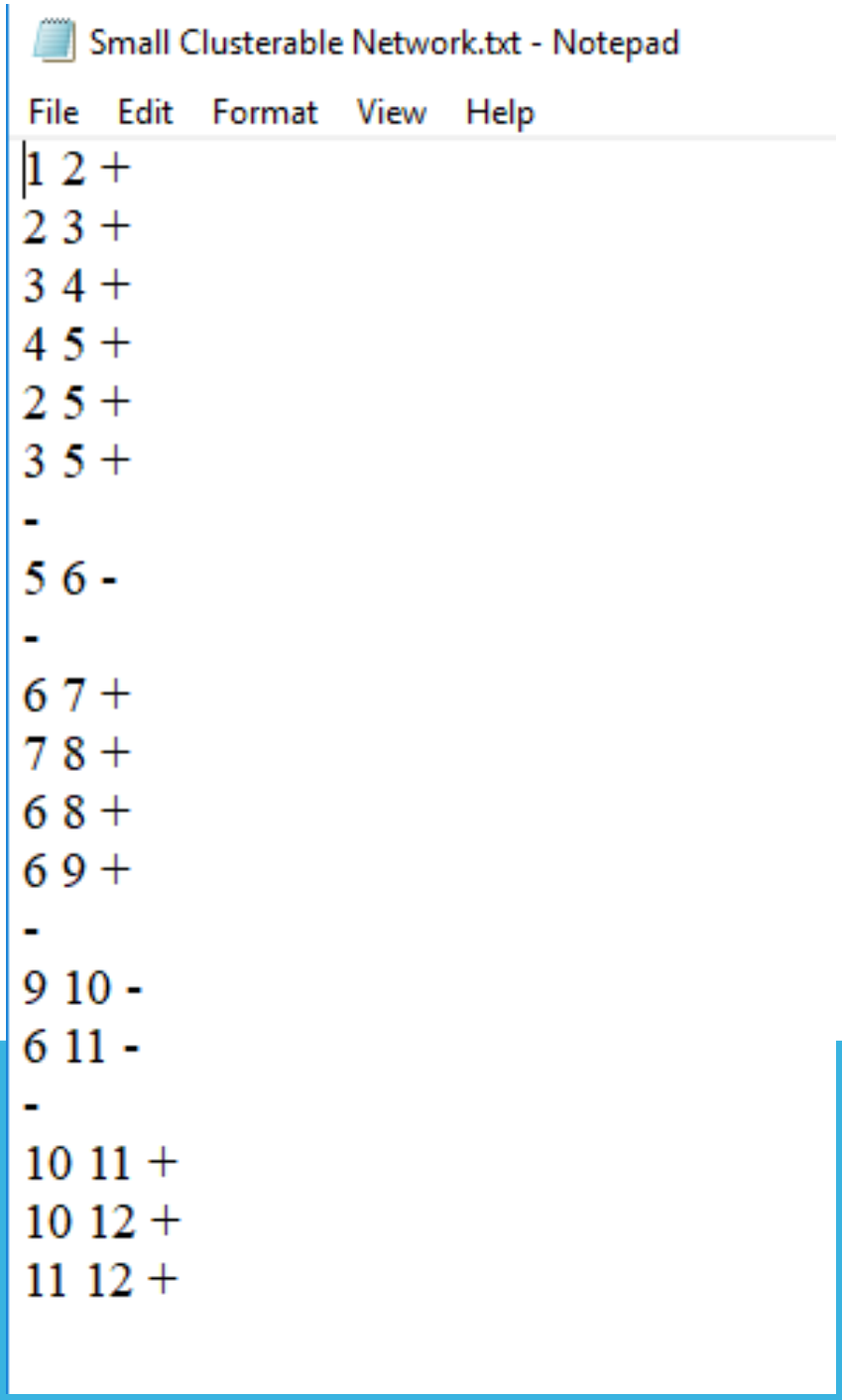
ПРОВЕРАВАЊЕ КОЛИКО КЛАСТЕРА У МРЕЖИ НИСУ КОАЛИЦИЈЕ

```
// CHECK HOW MANY CLUSTERS ARE NOT COALITIONS
int numberOfNOTCoalitionClusters = 0;
Collection<Set<Vertice>> NOTCoalitionClusters = clusterFinder.getNOTCoalitionClusterCollection();
writer.write("Clusters that ARE NOT coalitions clusters = " + System.lineSeparator());
writer.write(System.lineSeparator());

for (Set<Vertice> cluster : NOTCoalitionClusters) {
    if(cluster.size() < 2) {
        continue;
    }
    writer.write(cluster.toString() + " - (" + cluster.size() + " vertices)");
    writer.write(System.lineSeparator());
    numberOfNOTCoalitionClusters++;

    System.out.println("NOT COALITION = " + cluster);
}
writer.write("NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = " + numberOfNOTCoalitionClusters);
writer.write("-----------------------------------------------------------------------------");
writer.write(System.lineSeparator());
////////////////////////////////////////////////////////////////////////////////
```

ПРОВЕРАВАЊЕ КОЛИКО КЛАСТЕРА У МРЕЖИ СУ КОАЛИЦИЈЕ

```
// CHECK HOW MANY CLUSTERS ARE COALITIONS
clusters.removeAll(NOTCoalitionClusters);
int numberOfCoalitionClusters = 0;
writer.write("Clusters that ARE coalitions clusters = " + System.lineSeparator());
writer.write(System.lineSeparator());
for (Set<Vertice> cluster : clusters) {
    if(cluster.size() < 2) {
        continue;
    }
    writer.write(cluster.toString() + " - (" + cluster.size() + " vertices)");
    writer.write(System.lineSeparator());
    numberOfCoalitionClusters++;

    System.out.println("COALITION = " + cluster);
}
writer.write("NUMBER OF CLUSTERS THAT ARE COALITIONS = " + numberOfCoalitionClusters);
writer.write(System.lineSeparator());
writer.write(System.lineSeparator());
////////////////////////////////////////////////////////////////////////////////
```

ПРИМЕР МАЛЕ **КЛАСТЕРАБИЛНЕ** МРЕЖЕ



ПРИМЕР РЕЗУЛТАТА АНАЛИЗЕ МАЛЕ **КЛАСТЕРАБИЛНЕ** МРЕЖЕ

Small Clusterable Network.txt - Notepad

File Edit Format View Help

The network IS clusterable

CLUSTER 1(5 vertices) = Vertices:1,2,3,4,5Edges:+[2,5] +[3,5] +[2,3] +[3,4] +[1,2] +[4,5]

CLUSTER 2(4 vertices) = Vertices:6,7,8,9Edges:+[6,8] +[7,8] +[6,7] +[6,9]

CLUSTER 3(3 vertices) = Vertices:12,10,11Edges:+[10,12] +[11,12] +[10,11]

Clusters that ARE NOT coalitions clusters =

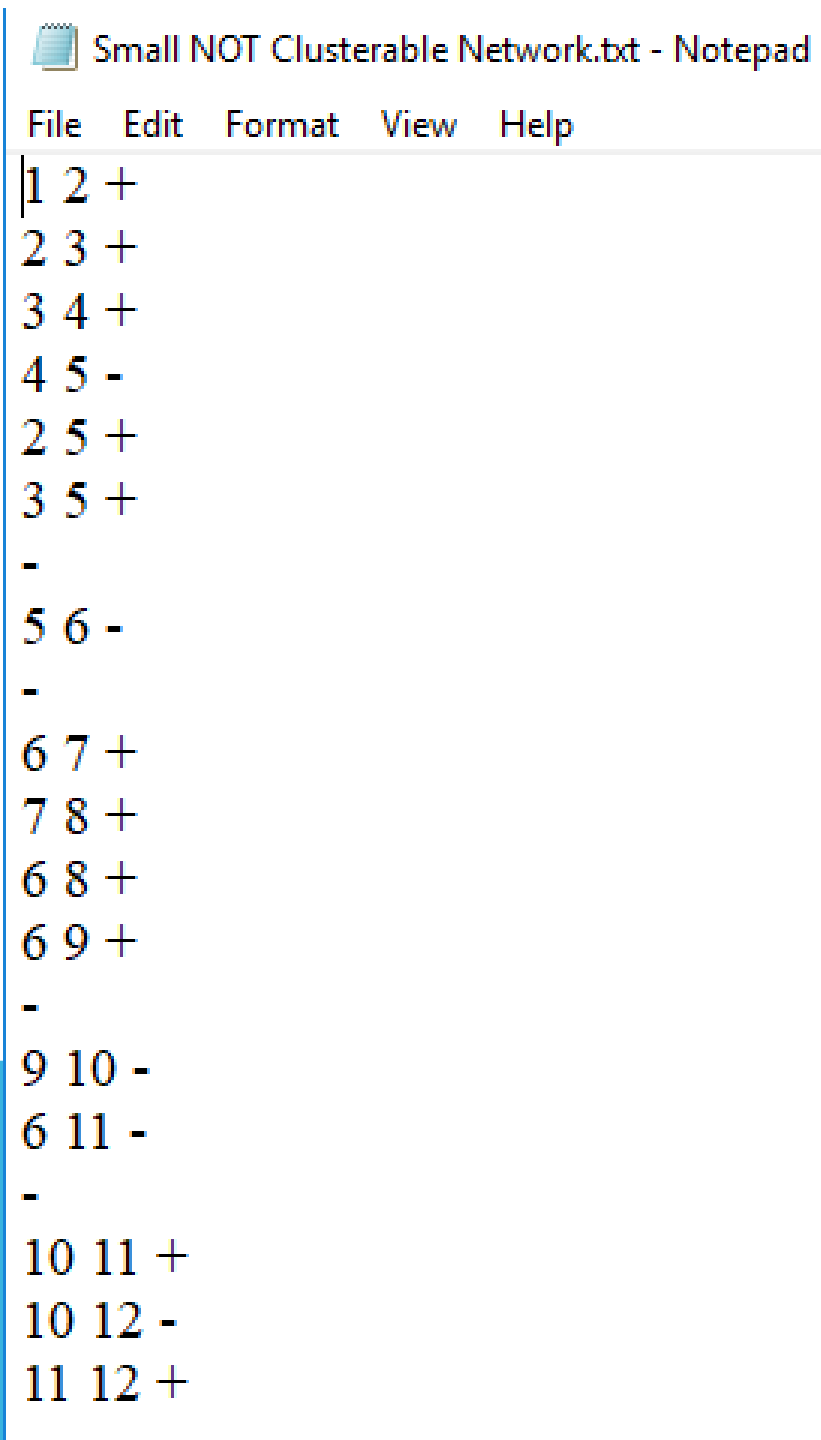
NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 0

Clusters that ARE coalitions clusters =

[1, 2, 3, 4, 5] - (5 vertices)
[6, 7, 8, 9] - (4 vertices)
[12, 10, 11] - (3 vertices)
NUMBER OF CLUSTERS THAT ARE COALITIONS = 3

PROGRAM TOOK = 0.0 SECONDS

ПРИМЕР МАЛЕ НЕКЛАСТЕРАБИЛНЕ МРЕЖЕ



ПРИМЕР РЕЗУЛТАТА АНАЛИЗЕ МАЛЕ **НЕКЛАСТЕРАБИЛНЕ** МРЕЖЕ

Small NOT Clusterable Network.txt - Notepad

File Edit Format View Help

Number of NODES in network = 12
Number of EDGES in network = 16

The network IS NOT clusterable

---

2 EDGES NEED TO BE DELETED
12.5% OF ALL EDGES NEED TO BE DELETED
IN ORDER FOR THE NETWORK TO BECOME CLUSTERABLE, THESE EDGES NEED TO BE DELETED =

-[10,12]
-[4,5]

CLUSTER 1(5 vertices) = Vertices:1,2,3,4,5Edges:+[2,5] +[3,5] +[2,3] +[3,4] +[1,2]

CLUSTER 2(4 vertices) = Vertices:6,7,8,9Edges:+[6,8] +[7,8] +[6,7] +[6,9]

CLUSTER 3(3 vertices) = Vertices:12,10,11Edges:+[11,12] +[10,11]

Clusters that ARE NOT coalitions clusters =

[1, 2, 3, 4, 5] - (5 vertices)
[12, 10, 11] - (3 vertices)
NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 2

Clusters that ARE coalitions clusters =

[6, 7, 8, 9] - (4 vertices)
NUMBER OF CLUSTERS THAT ARE COALITIONS = 1

PROGRAM TOOK = 0.0 SECONDS

ПРИМЕР РАНДОМ НЕКЛАСТЕРАБИЛНЕ МРЕЖЕ

RANDOM mreza.txt - Notepad

File Edit Format View Help

992 26 +
159 307 +
344 613 +
15 667 +
854 341 +
290 223 +
513 412 +
30 164 +
972 837 -
11 113 +
59 243 +
895 673 +
751 169 -
970 485 +
601 545 -
323 547 +
630 815 +
755 38 -
775 756 +
74 54 +
697 511 +

РЕЗУЛТАТИ АНАЛИЗЕ РАНДОМ **НЕКЛАСТЕРАБИЛНЕ** МРЕЖЕ

RANDOM mreza.txt - Notepad

File Edit Format View Help

Number of NODES in network = 1000
Number of EDGES in network = 3827

The network IS NOT clusterable

---

723 EDGES NEED TO BE DELETED
18.9% OF ALL EDGES NEED TO BE DELETED
IN ORDER FOR THE NETWORK TO BECOME CLUSTERABLE, THESE EDGES NEED TO BE DELETED =

-[276,455]
-[852,508]
-[778,915]
-[613,269]
-[614,550]
-[756,537]
-[828,676]
-[740,115]
-[568,802]
-[227,147]
-[922,668]
-[11,582]
-[198,653]
-[140,700]
-[201,970]
-[653,829]

- ЈЕДАН ВЕЛИКИ КЛАСТЕР КОЈИ НИЈЕ КОАЛИЦИЈА

Clusters that ARE NOT coalitions clusters =

[0, 1, 2, 3, 4, 5, 6, 7, 8, 800, 801, 9, 802, 803, 804, 805, 806, 807, 808,
844, 602, 845, 603, 604, 846, 847, 605, 848, 606, 849, 607, 608, 609, 8
869, 628, 629, 870, 871, 872, 630, 631, 873, 632, 874, 633, 875, 876, 6
408, 409, 890, 891, 892, 650, 893, 651, 410, 652, 411, 895, 653, 654, 4
670, 671, 430, 672, 673, 431, 674, 432, 433, 675, 676, 434, 677, 435, 4
691, 692, 450, 451, 693, 210, 452, 694, 211, 453, 695, 696, 212, 454, 4
230, 472, 231, 473, 474, 232, 475, 233, 234, 476, 235, 477, 236, 478, 4
255, 256, 498, 499, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 2
904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 9
948, 707, 949, 708, 709, 950, 951, 710, 952, 953, 711, 954, 712, 955, 7
974, 732, 975, 733, 976, 734, 977, 735, 736, 978, 737, 979, 738, 739, 9
510, 752, 753, 995, 511, 512, 754, 996, 755, 513, 997, 756, 514, 998, 5
775, 533, 776, 534, 535, 777, 778, 536, 779, 537, 538, 539, 780, 781, 7
553, 311, 795, 312, 554, 796, 797, 313, 555, 314, 556, 798, 799, 557, 3
576, 334, 335, 577, 336, 578, 337, 579, 338, 339, 580, 581, 340, 582, 5
595, 354, 596, 112, 355, 113, 597, 114, 598, 356, 357, 599, 115, 358, 1
27, 28, 29, 370, 371, 130, 372, 131, 373, 374, 132, 375, 133, 376, 134,
41, 42, 43, 44, 45, 46, 47, 48, 49, 390, 391, 150, 392, 393, 151, 152, 3
69, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 70, 71, 72, 73, 74
97, 98, 99] - (998 vertices)

NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 1

Clusters that ARE coalitions clusters =

NUMBER OF CLUSTERS THAT ARE COALITIONS = 0

PROGRAM TOOK = 0.4 SECONDS

ПРИМЕР РАНДОМ **КЛАСТЕРАБИЛНЕ** МРЕЖЕ

RANDOM mreza.txt - Notepad

File Edit Format View Help

1067 678 +
65 965 +
269 1297 -
1089 147 -
781 1306 +
1107 1195 +
1197 580 +
233 566 +
810 1096 +
587 637 +
261 949 -
1233 804 +
804 667 -
225 590 +
901 513 +
914 351 -
777 162 +
19 57 -
411 951 +
915 969 +
956 1246 +
1296 900 -
1388 520 +

РЕЗУЛТАТИ АНАЛИЗЕ РАНДОМ **КЛАСТЕРАБИЛНЕ** МРЕЖЕ

RANDOM mreza.txt - Notepad

File Edit Format View Help

Number of NODES in network = 1000
Number of EDGES in network = 847

The network IS clusterable

CLUSTER 1(4 vertices) = Vertices:1347,545,777,162Edges:+[545,1347] +[777,162] +[545,777]

CLUSTER 2(2 vertices) = Vertices:810,1096Edges:+[810,1096]

CLUSTER 3(11 vertices) = Vertices:794,366,433,137,874,216,623,636,1033,1063,818Edges:+[10
+[1033,623] +[1033,216] +[1063,216]

CLUSTER 4(2 vertices) = Vertices:325,118Edges:+[325,118]

CLUSTER 5(5 vertices) = Vertices:772,776,923,1163,680Edges:+[680,1163] +[680,776] +[923,11

CLUSTER 6(2 vertices) = Vertices:188,1165Edges:+[188,1165]

CLUSTER 7(2 vertices) = Vertices:578,1345Edges:+[578,1345]

CLUSTER 8(5 vertices) = Vertices:474,114,349,1396,1141Edges:+[114,1141] +[114,474] +[349,1

CLUSTER 9(8 vertices) = Vertices:750,400,1127,621,9,849,1051,790Edges:+[750,1051] +[790,84

CLUSTER 10(2 vertices) = Vertices:411,951Edges:+[411,951]

CLUSTER 11(27 vertices) = Vertices:919,1012,1273,1196,1074,584,982,961,149,567,525,108,966
+[919,512] +[149,1143] +[505,1119] +[919,791] +[42,831] +[831,966] +[961,150] +[215,1273] +[

ЈОШ НЕКИ ОД КЛАСТЕРА У МРЕЖИ

CLUSTER 95(2 vertices) = Vertices:1368,370Edges:+[1368,370]

CLUSTER 96(9 vertices) = Vertices:1028,587,1235,1386,637,615,1197,580,1161Edges:+[1028,587] +[587,637] +[637,1161
+[1197,1386]

CLUSTER 97(2 vertices) = Vertices:926,829Edges:+[926,829]

CLUSTER 98(3 vertices) = Vertices:1137,677,261Edges:+[677,1137] +[677,261]

CLUSTER 99(98 vertices) =
Vertices:1331,1293,1050,1170,221,620,224,742,502,80,626,909,1066,1186,350,752,115,753,1213,479,875,755,119,912,1219
4,250,495,651,377,894,1359,499,538,819,1010,140,263,266,1006,424,303,788,822,703,307,1020,554,1258,713,438,59,1034
9,728,607,170,1284,1162,570,1282,210,574,971,455,457,458,975,73,738Edges:+[1293,1258] +[455,1213] +[620,1010] +[97
+[350,221] +[703,73] +[455,1268] +[562,502] +[1269,753] +[894,479] +[24,224] +[499,894] +[971,1219] +[424,742] +[75
+[140,1066] +[570,17] +[713,484] +[1010,221] +[755,964] +[115,457] +[243,1268] +[753,768] +[802,323] +[406,307] +[1
+[527,119] +[160,752] +[1162,755] +[484,1170] +[894,1186] +[912,210] +[921,1258] +[822,119] +[1282,788] +[1304,975]
+[458,806] +[160,21] +[570,912] +[538,822] +[875,406] +[484,170] +[1020,1304] +[80,160] +[809,21] +[458,1050] +[526
+[303,115] +[1050,250] +[651,221] +[806,554] +[495,221] +[1269,738] +[209,457] +[1331,640] +[975,323] +[806,499] +[
+[170,303] +[377,307] +[1268,250] +[324,502] +[640,912] +[527,209] +[554,263] +[1186,819] +[1359,266] +[1010,626] +

CLUSTER 100(2 vertices) = Vertices:226,295Edges:+[226,295]

CLUSTER 101(2 vertices) = Vertices:117,165Edges:+[117,165]

CLUSTER 102(5 vertices) = Vertices:1306,1155,1110,274,781Edges:+[781,1306] +[1110,1155] +[274,1155] +[781,1110]

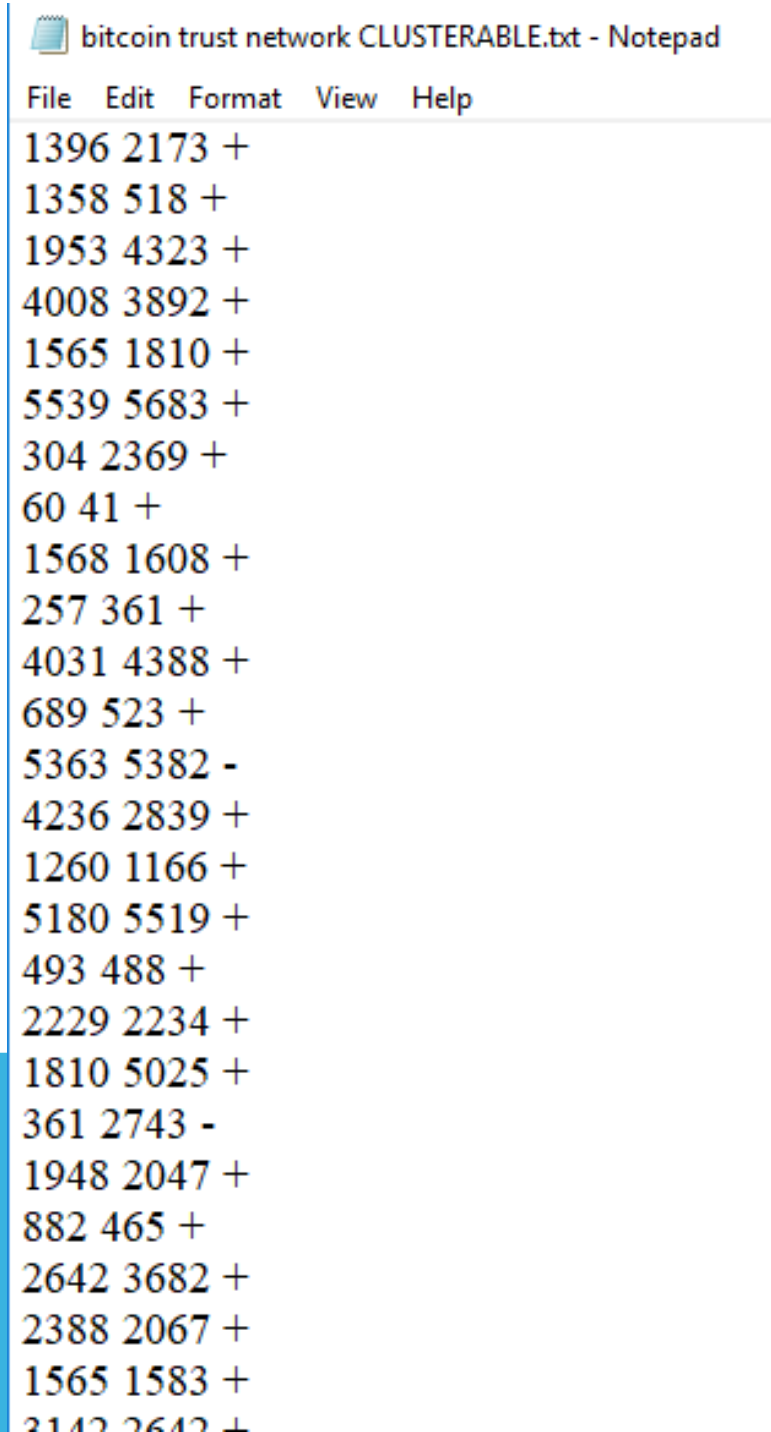
CLUSTER 103(2 vertices) = Vertices:312,887Edges:+[312,887]

- ВЕЛИКИ БРОЈ КЛАСТЕРА КОЈИ СУ КОАЛИЦИЈЕ
- МРЕЖА ЈЕ **КЛАСТЕРАБИЛНА**

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Clusters that ARE NOT coalitions clusters =

NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 0
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
```

```
NUMBER OF CLUSTERS THAT ARE COALITIONS = 206

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
PROGRAM TOOK = 0.1 SECONDS
```

ПРИМЕР МРЕЖЕ ПОВЕРЕЊА МЕЂУ КОРИСНИЦИМА БИТКОИНА



- ЈЕДАН ГИГАНТСКИ КЛАСТЕР И НЕКОЛИКО МАЊИХ

bitcoin trust network CLUSTERABLE.txt - Notepad

File Edit Format View Help

Number of NODES in network = 5881
Number of EDGES in network = 19405

The network IS clusterable

CLUSTER 1(5550 vertices) =
Vertices:4960,3630,4961,3628,4959,3629,36
34,4966,3635,4967,3636,4968,2309,3637,49
90,4991,3660,4992,3661,4993,3662,4994,36
09,1008,2339,1007,2337,1006,1005,2336,10
16,2346,1015,1014,2345,2344,1013,3675,36
44,2375,1043,2374,1042,2373,1041,2372,10
77,1046,5403,5404,5408,5409,2390,5420,54
78,1077,1076,1075,1074,1073,1072,1071,10
9,4108,4109,1092,5450,1091,5451,4120,109
,4128,5459,4129,4140,5472,4141,5473,4142
150,210,211,212,213,214,215,216,217,5479

+[4172,2767] +[1953,5404] +[2028,2576] +[2063,1915] +[1754,1912] +[1528,23
+[4861,4559] +[1885,230] +[3449,2388] +[832,873] +[1053,522] +[35,5816] +[2
+[2028,1731] +[3653,2266] +[7,134] +[1162,1555] +[3878,5525] +[1396,1948] +
+[2249,2414] +[3129,4069] +[3935,1976] +[2141,2232] +[1894,1921] +[616,399
+[1018,2963] +[320,60] +[35,3587] +[467,463] +[905,1528] +[3578,4360] +[110
+[4172,3854] +[547,687] +[353,1565] +[60,243] +[1864,523] +[5312,5345] +[34
+[35,4067] +[2963,1953] +[35,1440] +[4074,3513] +[2135,2349] +[4172,4788] +
+[3722,1512] +[1281,2094] +[28,2618] +[1543,1400] +[3987,3598] +[760,834] +
+[1512,5525] +[2045,5917] +[1115,1348] +[2800,2763] +[1629,2192] +[1802,229

CLUSTER 2(2 vertices) = Vertices:1327,1329Edges:+[1329,1327]

CLUSTER 3(2 vertices) = Vertices:3762,3763Edges:+[3762,3763]

CLUSTER 4(2 vertices) = Vertices:3911,3912Edges:+[3911,3912]

CLUSTER 5(2 vertices) = Vertices:6000,6002Edges:+[6000,6002]

CLUSTER 6(2 vertices) = Vertices:4741,4742Edges:+[4741,4742]

CLUSTER 7(2 vertices) = Vertices:1600,1572Edges:+[1572,1600]

CLUSTER 8(2 vertices) = Vertices:5471,5544Edges:+[5471,5544]

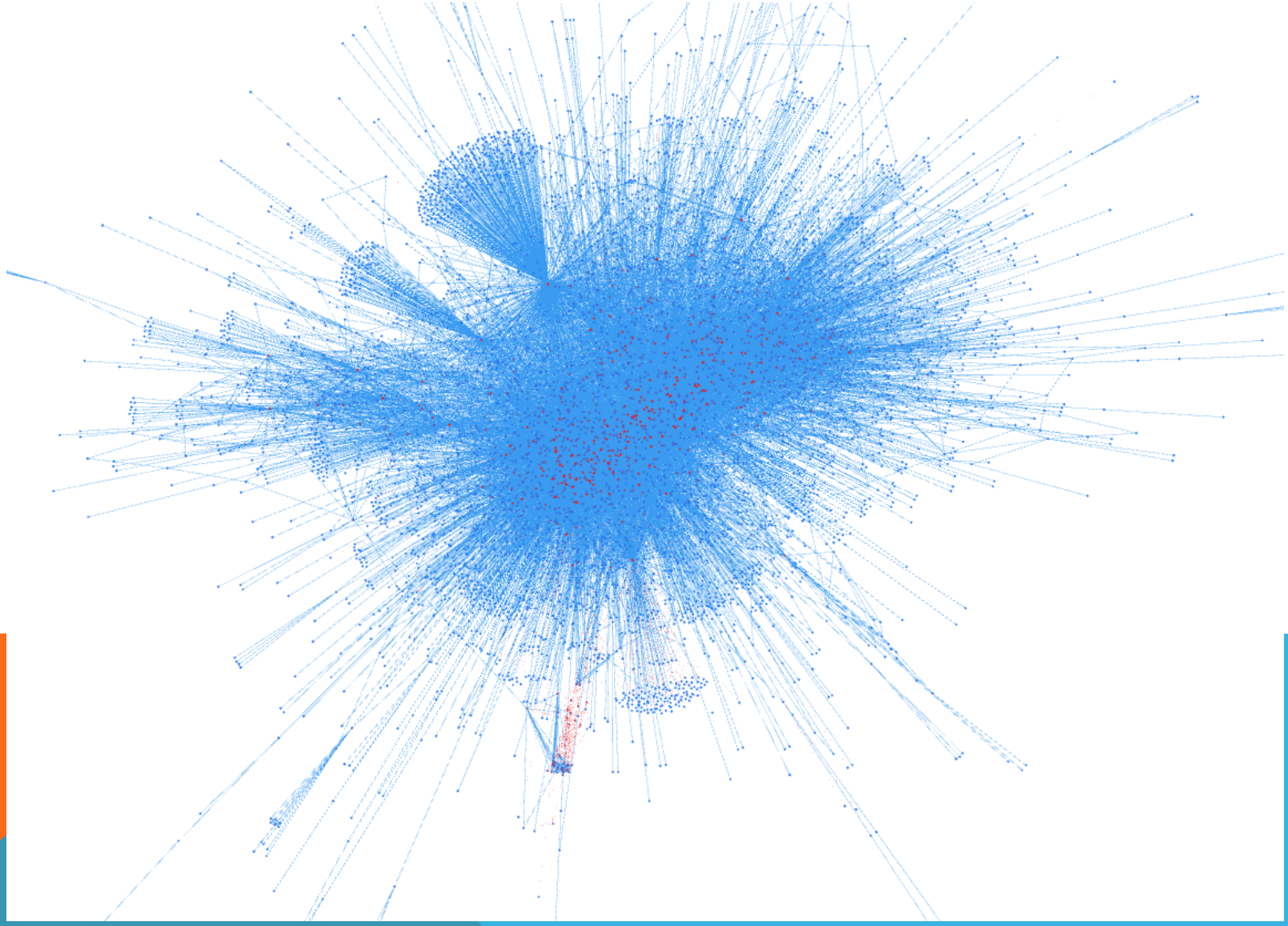
CLUSTER 9(8 vertices) = Vertices:5197,5198,5200,5201,5202,5203,5204,5193Ec

---

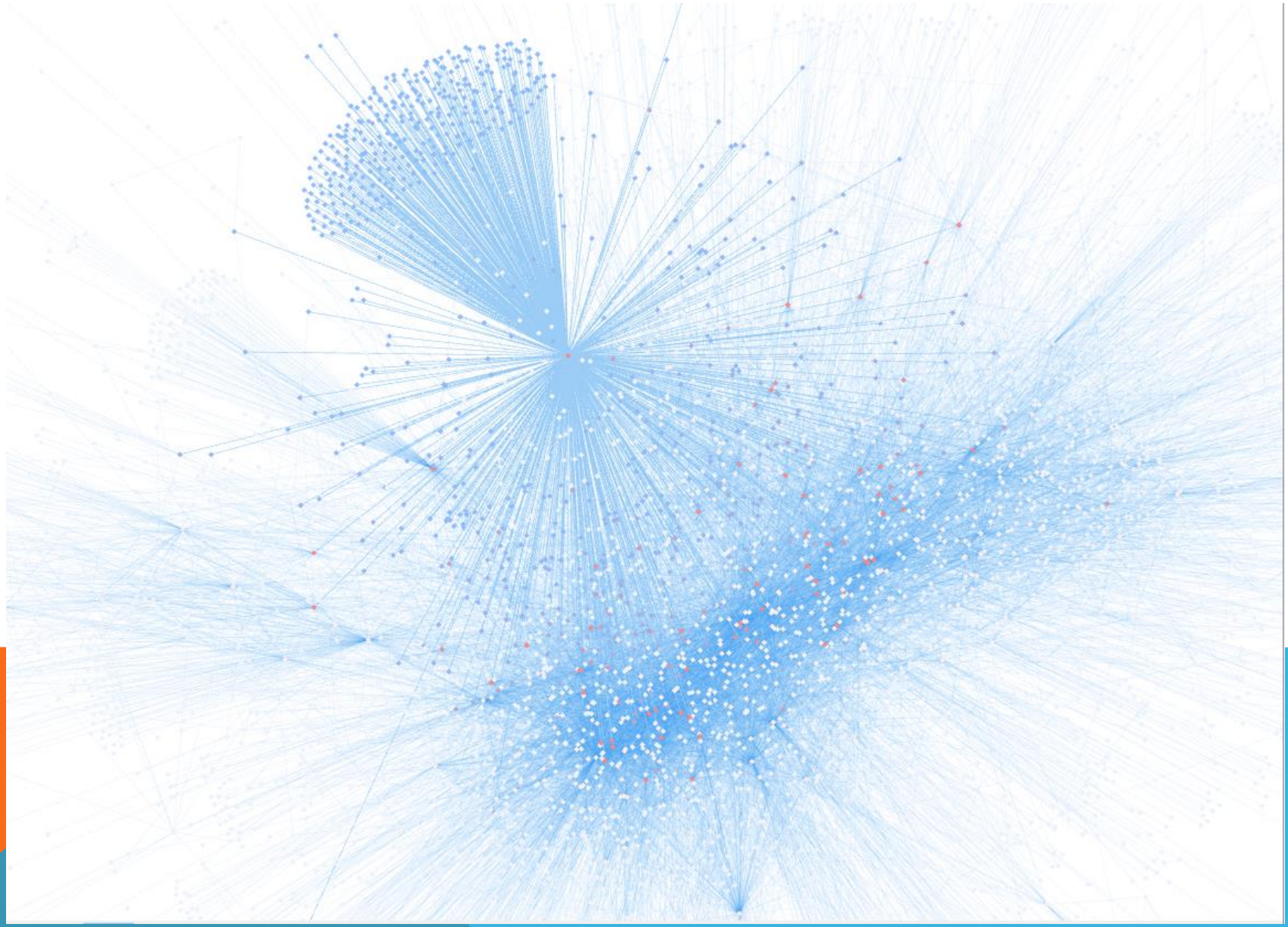
Clusters that ARE NOT coalitions clusters =

NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 0

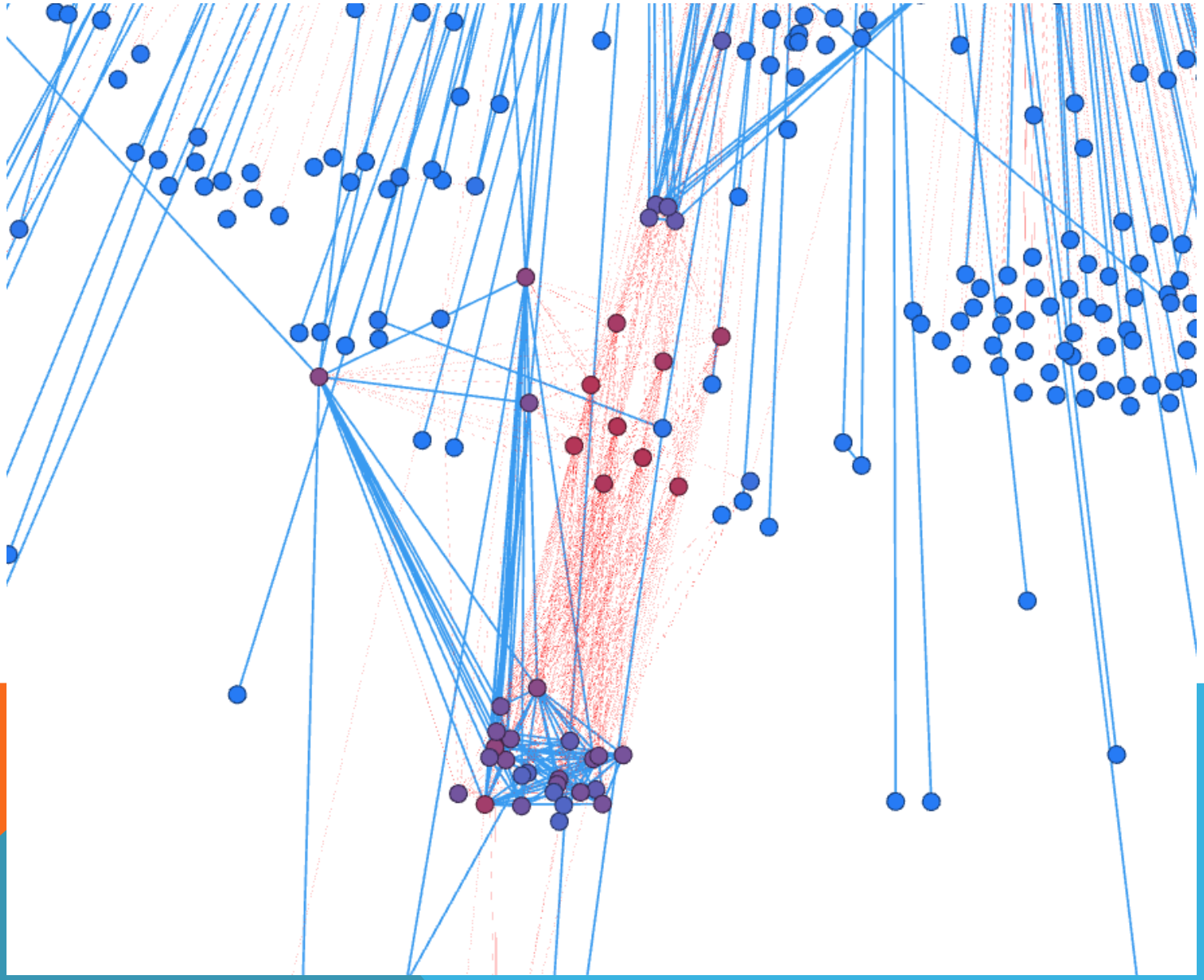
МРЕЖА ПОВЕРЕЊА БИТКОИН КОРИСНИКА



ЈЕДАН ОД ХАБОВА У МРЕЖИ



ПОДМРЕЖА ХЕЈТЕРА



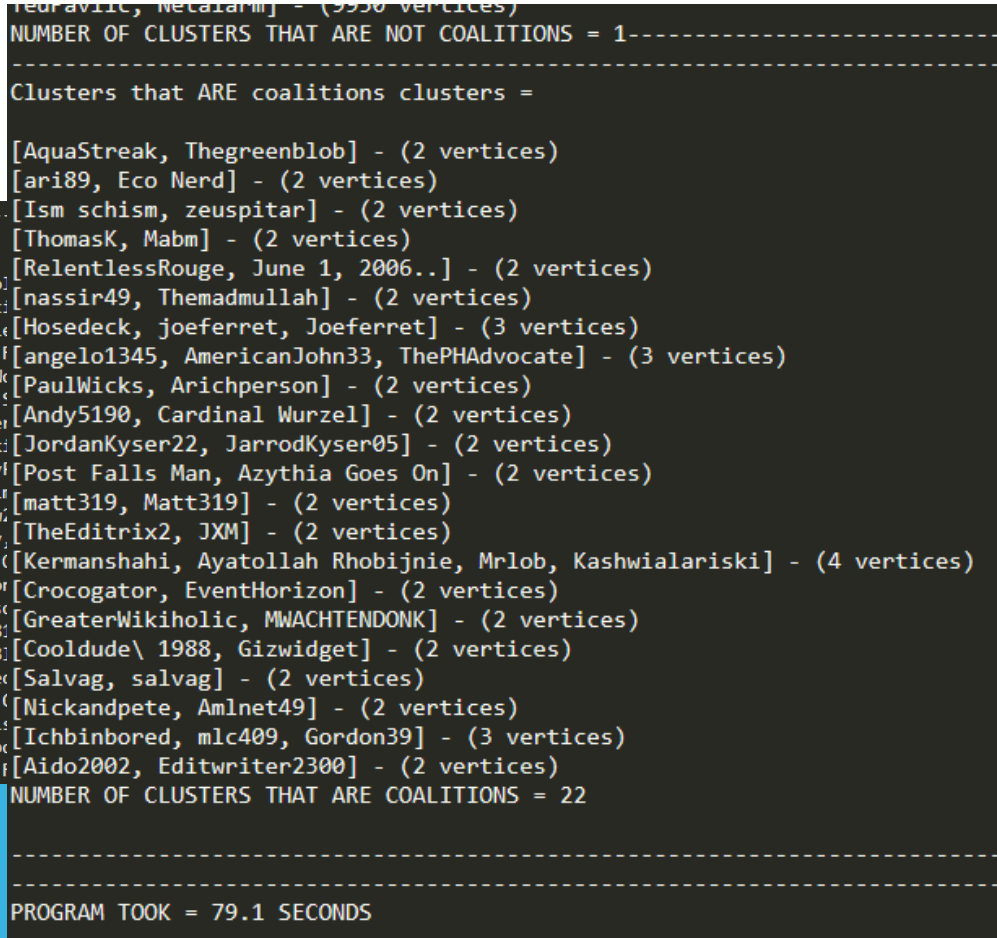
ПРИМЕР МРЕЖЕ КОРИСНИКА ВИКИПЕДИЈЕ И ЊИХОВА ПОДРШКА ИЛИ ОДБИЈАЊЕ ЗАХТЕВА ЗА ДОБИЈАЊЕ АДМИНА

```
205593  SRC:Collect
205594  TGT:Sphilbrick
205595  VOT:1
205596  RES:1
205597  YEA:2010
205598  DAT:00:45, 15 November 2010
205599  TXT:'''Support''' Noting that original thoughts are actually found in the answers rather than just iterating
205600
205601  SRC:Townlake
205602  TGT:Sphilbrick
205603  VOT:1
205604  RES:1
205605  YEA:2010
205606  DAT:02:38, 15 November 2010
205607  TXT:I frequently oppose candidates for being drama magnets. However, this particular candidate manages drama
        weirdness within and around it. Happy to support; good luck and happy admin-ing.
205608
205609  SRC:Regent of the Seatopians
205610  TGT:Sphilbrick
205611  VOT:1
205612  RES:1
205613  YEA:2010
205614  DAT:03:25, 15 November 2010
205615  TXT:'''Support''' A good choice for adminship.
205616
205617  SRC:Banana04131
205618  TGT:Sphilbrick
205619  VOT:1
205620  RES:1
205621  YEA:2010
205622  DAT:04:42, 15 November 2010
205623  TXT:'''Support''' Nice answers to questions. --
```

РЕЗУЛТАТИ АНАЛИЗЕ МРЕЖЕ

```
Number of NODES in network = 11370
Number of EDGES in network = 181026

The network IS NOT clusterable
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
44075 EDGES NEED TO BE DELETED
24.3% OF ALL EDGES NEED TO BE DELETED
IN ORDER FOR THE NETWORK TO BECOME CLUSTERABLE, THESE EDGES NEED TO BE DELETED =

-[Jusjih,Srikeit]
-[Thumbelina,Werdna648]
-[Neurolysis,Graymornings]
-[Mailer diablo,Mac Davis]
-[X!,RyRy]
-[Alex Bakharev,Chacor]
-[GiantSnowman,Miniapolis]
-[Cyrius,IndigoGenius]
-[Tone,M1ss1ontomars2k4]
-[Elkman,ACBest]
-[Kbdank71,MONGO]
-[Majorly,Moreschi]
-[Sir Nicholas de Mimsy-Porpington,Everyking]
-[Dank,Download]
-[Aqwis,Nosleep]
-[Epeefleche,MZMcBride]
-[Gordonrox24,Apteva]
-[Danny,ChrisDJackson.09]
-[Whereizben,Sox23]
-[Kimchi.sg,Schzmo2]
-[Arf!,Hdt83]
-[Shreshth91,Dustimagic]
-[Off2riorob,Acdixon]
-[Allstarecho,ChildofMidnight]
-[Balloonman,Sephiroth storm]
-[Fastily,TedPavlic]
-[Fabrib,Markovich292]
-[RyRy,Danielfolsom]
-[Pablothegreat85,Aktron]
-[Mike1,Thatcher131]
-[Malinaccier,Mvjs]
-[Imageboy1,Nima Baghaei]
```

РЕЗУЛТАТИ АНАЛИЗЕ МРЕЖЕ



ПРИМЕР МРЕЖЕ КОРИСНИКА ВИКИПЕДИЈЕ И ЊИХОВА ПОДРШКА ИЛИ ОДБИЈАЊЕ ЗАХТЕВА ЗА ДОБИЈАЊЕ АДМИНА
- НАПРАВЉЕНА ТАКО ДА ЈЕ КЛАСТЕРАБИЛНА

PookeyMaster Masem +
Zsinj Philosopher +
Khoikhoi Biruitorul +
Miranda Keeper76 +
CactusWriter Aervanath +
Carlosguitar Oxymoron83 +
SoWhy Mizu -
Draicone Eluchil404 +
Coemgenus Canadian-Bacon +
Hiberniantears Fribbulus -
lovelaughterlife Jersyko +
B Werdna +
Kubigula Looie496 +
Ret.Prof Tinucherian +
Sceptre AKMask +
Angusmclellan Berig +
Peripitus Finalnight +
Ugen64 Wilfried -
PhilKnight Mandsford +
Titoxd Natalya +
Getcrunk Dewet +
Tempodivalse Kingpin13 +
Dorftrottel SkierRMH +
Mixwell FlyingToaster +
LAX Ultraexactzz +
RP459 Katerenka +
Vertium Î£ +
Slgrandson Soap +
Reddi William -
Brian0918 Nickptar +
Beeblebrox Tim -
Keepscases My76Strat +
Dwaipayanc Fuhghettaboutit +

РЕЗУЛТАТИ АНАЛИЗЕ МРЕЖЕ

```
The network IS clusterable
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
CLUSTER 1(2 vertices) = Vertices:AquaStreak,Thegreenblob
Edges:+[Thegreenblob,AquaStreak]

CLUSTER 2(2 vertices) = Vertices:nassir49,Themadmullah
Edges:+[Themadmullah,nassir49]

CLUSTER 3(2 vertices) = Vertices:PaulWicks,Arichperson
Edges:+[Arichperson,PaulWicks]

CLUSTER 4(2 vertices) = Vertices:matt319,Matt319
Edges:+[Matt319,matt319]

CLUSTER 5(2 vertices) = Vertices:Salvag,salvag
Edges:+[Salvag,salvag]

CLUSTER 6(2 vertices) = Vertices:JustPhil,Myrtone86
Edges:+[Myrtone86,JustPhil]

CLUSTER 7(2 vertices) = Vertices:TheEditrix2,JXM
Edges:+[TheEditrix2,JXM]

CLUSTER 8(7533 vertices) = Vertices:Sesel,Mtking,tszho1997,RadManCF,Bouncingmolar,St.daniel,Steele,Wolfma
ulevardier,SD5,Alvestrand,Dudemanfellabra,Ihcoyc,Froggydarb,Daven200520,BuddingJournalist,Cjmarsicano,Kor
Oakland,2,Nauticashades,7,Satellizer,Polonium,Ronbo76,Defrosted,Rambo,Edgar181,Naerii,BDD,A,B,Abd,Rama,E,
,Oggleboppiter,j,Genius101,SwirlBoy39,Libertyville,elcobbola,v,MikeHunt35,Diez2,SluggoOne,Maxamegalon2000
Fallstorm,MathCool10,Cassivs,Rfl,Ahunt,Adz,Stevietheman,Herostratus,OwenBlacker,Kimon,Chuckfromchan,KojiD
ndawydiak,Orion11M87,Blurpeace,Grue,FunkyFly,Sharkface217,Vsion,Ford,Xyrael,CBDunkerson,x›x³x¯,Utopianhea
krus,Bullzeye,SJP,CrispMuncher,Sjakkalle,Cailil,Bobthefish2,Shappy,Andy4789,Jack-A-Roe,Hamiltonstone,MSGJ
nzertank,Mattopaedia,Aitias,Karmosin,Rje,Jyril,Richwales,Bobo192,CaSJer,JavaTenor,Caden,Karimarie,Kurtis,
perSushi21,Pdcook,SMC,Butseriouslyfolks,Verbal,ciphergoth,DragonRouge,Coldmachine,Spell4yr,Jacob696,Singu
ard,Lucasbunchi,ArielGold,Dominic,Crisspy,Decker41811a,roux,C.Fred,Threeafterthree,Anecdote,Jergen,Angela
haitza,Cunya,unforgettableid,Eagleamn,Omniplex,Walton77,Zanaq,Yacht,MikeBeckett,Anthon01,Comic,Semifreddo
enine,truthspreader,Venomcuz,beneaththelandslide,Netesq,Extransit,GenghizRat,LittleOldMe,Chairboy,Mfield,
i,Callipides,Redmarkviolinist,SupaSoldier,DavidJJJ,Naconkantari,LedgendGamer,Nikki311,Alcidebava,Gobonobo
a,Giovanni33,Mdcollins1984,IndigoGenius,JamesTeterenko,Veesicle,PopularMax,Centrx,BorgQueen,Ravedave,Xtra
donic,Mattbr,MatthewUND,dbertman,Yoshaibo,Seav,Itai.09,Stemonitis,Abhijay,Bongomatic,DoriSmith,Adoptlesds
,danntm,MasterEagle,DCUnitedFan2011,Eahiv,Wikidudeman,Maltesedog3,imdanumber1,Diego,Jackieboy87,Rossami,A
rk30inf,Aua,BWD,Someone65,Frozen4322,Peterwats,Moreschi,Unioneagle,Cheeser1,Odysses,RevRagnarok,Jrphayes,
tu,Luckytoilet,Dspradau,ShakespeareFan00,JimMillerJr,Markalexander100.09,B.hotep,AxG,Jahangard,Epipelagic
```

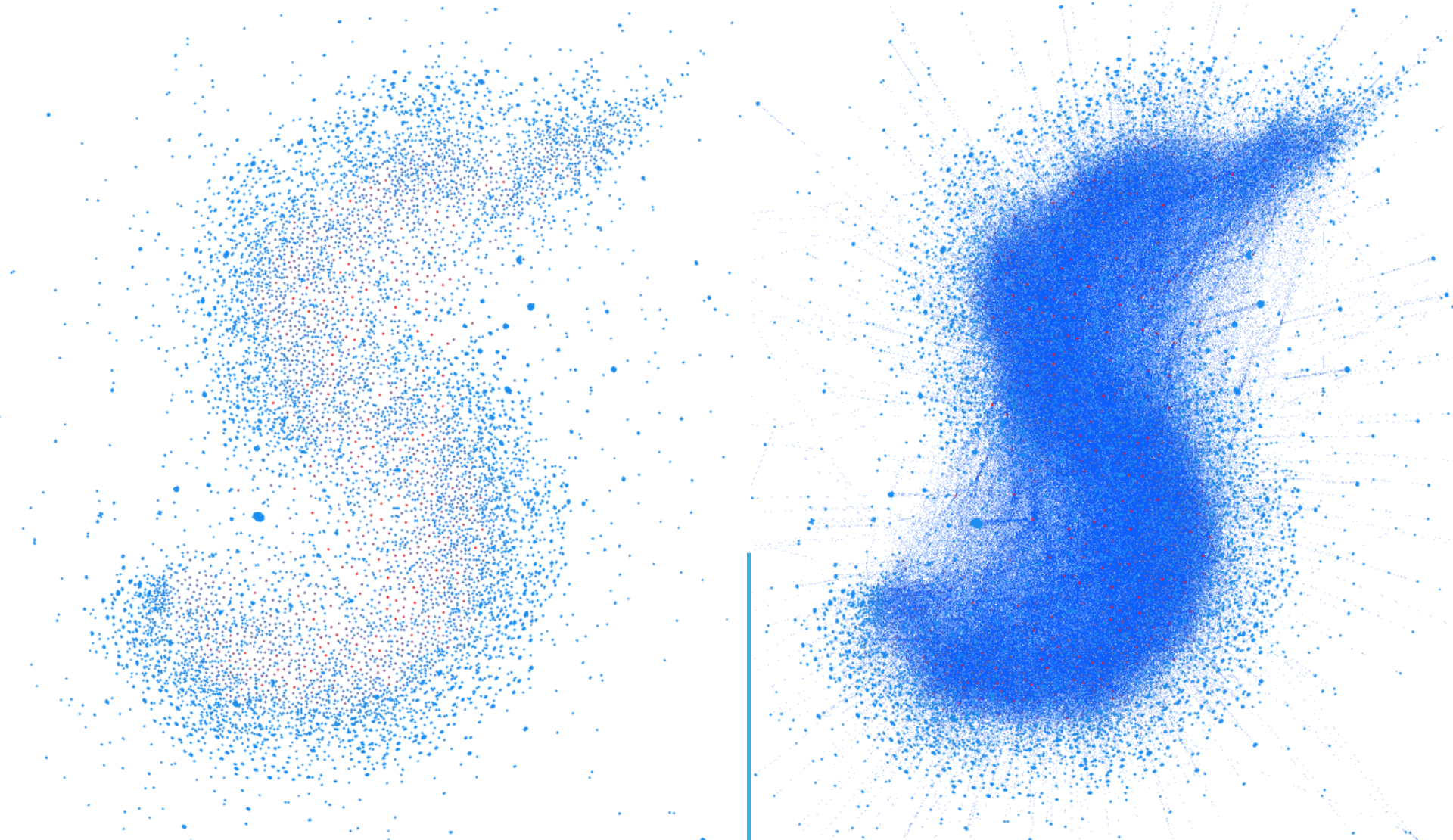
```
Clusters that ARE NOT coalitions clusters =

NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 0---
```

```
Hoopydink, Tmorton166, Masterpiece2000, Nat91, Rustavo, JWSc
Bastun, Sparkit, Pablo-flores, Goobergunch, HolyRomanEmperor
Billreid, GregAsche, Plek, Mattinbgn, Huon, Immunize, JimmyB
Catamorphism, Dcollins52, Member, Mail2amitabha, Butterflys,
[Kermanshahi, Mrlob, Kashwialariski] - (3 vertices)
[ThomasK, Mabm] - (2 vertices)
[Hosedeck, joeferret, Joeferret] - (3 vertices)
[angelo1345, AmericanJohn33, ThePHAdvocate] - (3 vertices)
[Nickandpete, Amlnet49] - (2 vertices)
[Ichbinbored, mlc409, Gordon39] - (3 vertices)
[Aido2002, Editwriter2300] - (2 vertices)
[Mrbowtie, Jareand] - (2 vertices)
[JordanKyser22, JarrodKyser05] - (2 vertices)
[Crocogator, EventHorizon] - (2 vertices)
[GreaterWikiholic, MWACHTENDONK] - (2 vertices)
[Lambertman, Kramden4700] - (2 vertices)
NUMBER OF CLUSTERS THAT ARE COALITIONS = 20

------------------------------------------------------------
------------------------------------------------------------
PROGRAM TOOK = 12.9 SECONDS
```

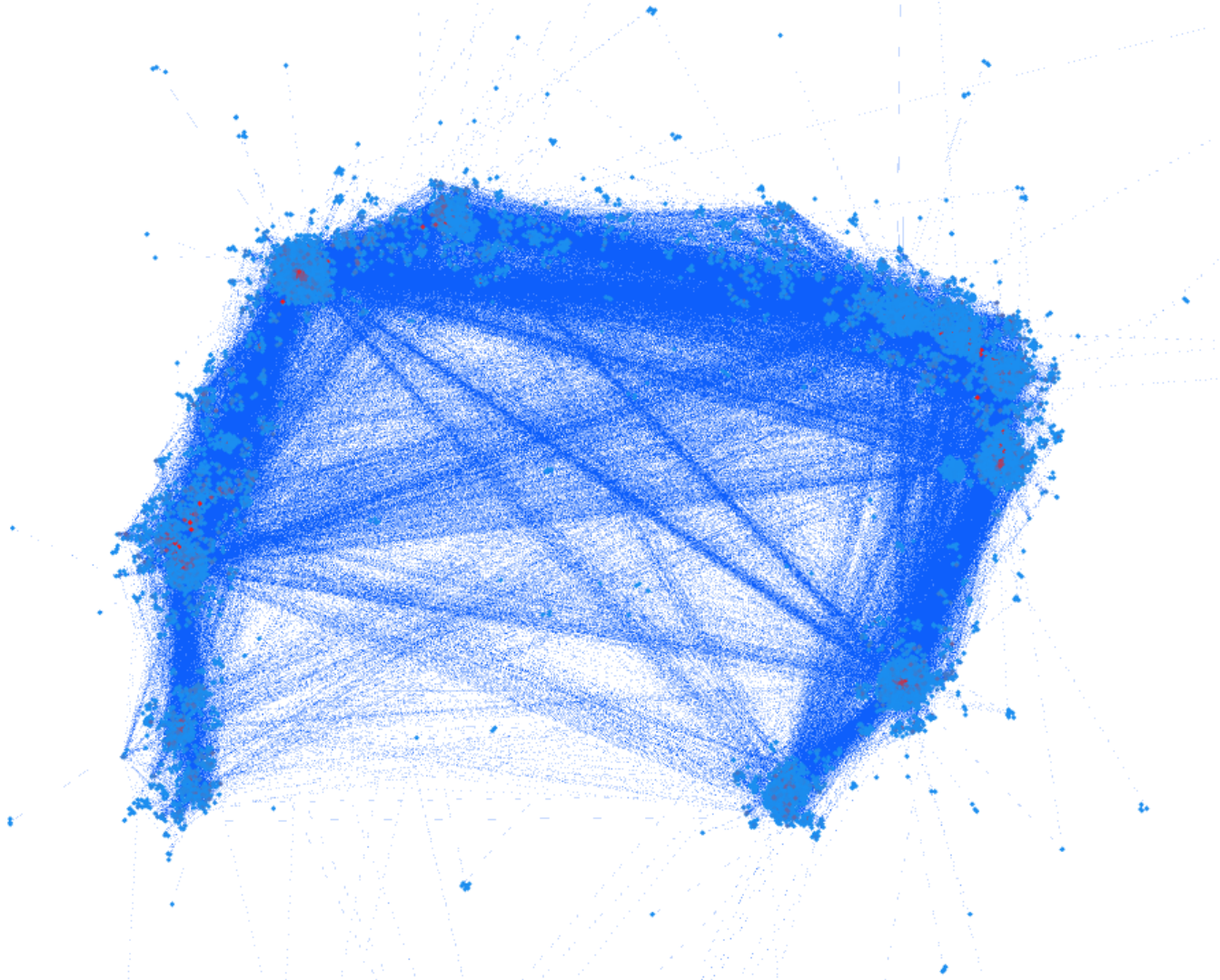
ВИЗУАЛИЗАЦИЈА МРЕЖЕ

САМО ТАЧКЕ

ТАЧКЕ СА ВЕЗАМА



ЈОШ ЈЕДАН НАЧИН ВИЗУАЛИЗАЦИЈЕ



ПРИМЕР МРЕЖЕ ОДНОСА ИЗМЕЂУ РАЗНИХ REDDIT СТРАНИЦА

```
leagueoflegends teamredditteams 1
theredlion  soccer  -1
inlandempire    bikela  1
nfl cfb 1
playmygame  gamedev 1
dogemarket  dogecoin    1
locationbot legaladvice 1
indiefied   aww 1
posthardcore    bestof2013  1
posthardcore    corejerk    1
gfycat  india   1
metalcore   bestof2013  1
metalcore   corejerk    1
suicidewatch    offmychest  1
dogecoin    novacoin    1
gaming4gamers   fallout 1
kpop    dota2   1
airsoft airsoftmarket   1
circlebroke childfree   1
tribes  games   1
oldschoolcoolnsfw   pics    1
fl_vapers   vaperequests    1
jailbreak   flextweak   1
corejerk    bestof2013  1
iama    todayilearned   1
bandnames   books   1
thedoctorstravels   hungergamesrp   1
politicaldiscussion todayilearned   1
uncomfortableqs debatereligion  1
connecticut ctbeer  1
metafitnesscirclejerk   fitnesscirclejerk   1
srssucks    funny   1
thehiddenbar    writingprompts  1
circlejerkcopypasta askreddit   1
karmaconspiracy funny   -1
denverbroncos   seahawks    1
askreddit   todayilearned   1
civcraft    leningrad   1
gaybros askreddit   1
tupptablists    vinyl   1
```

РЕЗУЛТАТИ АНАЛИЗЕ МРЕЖЕ

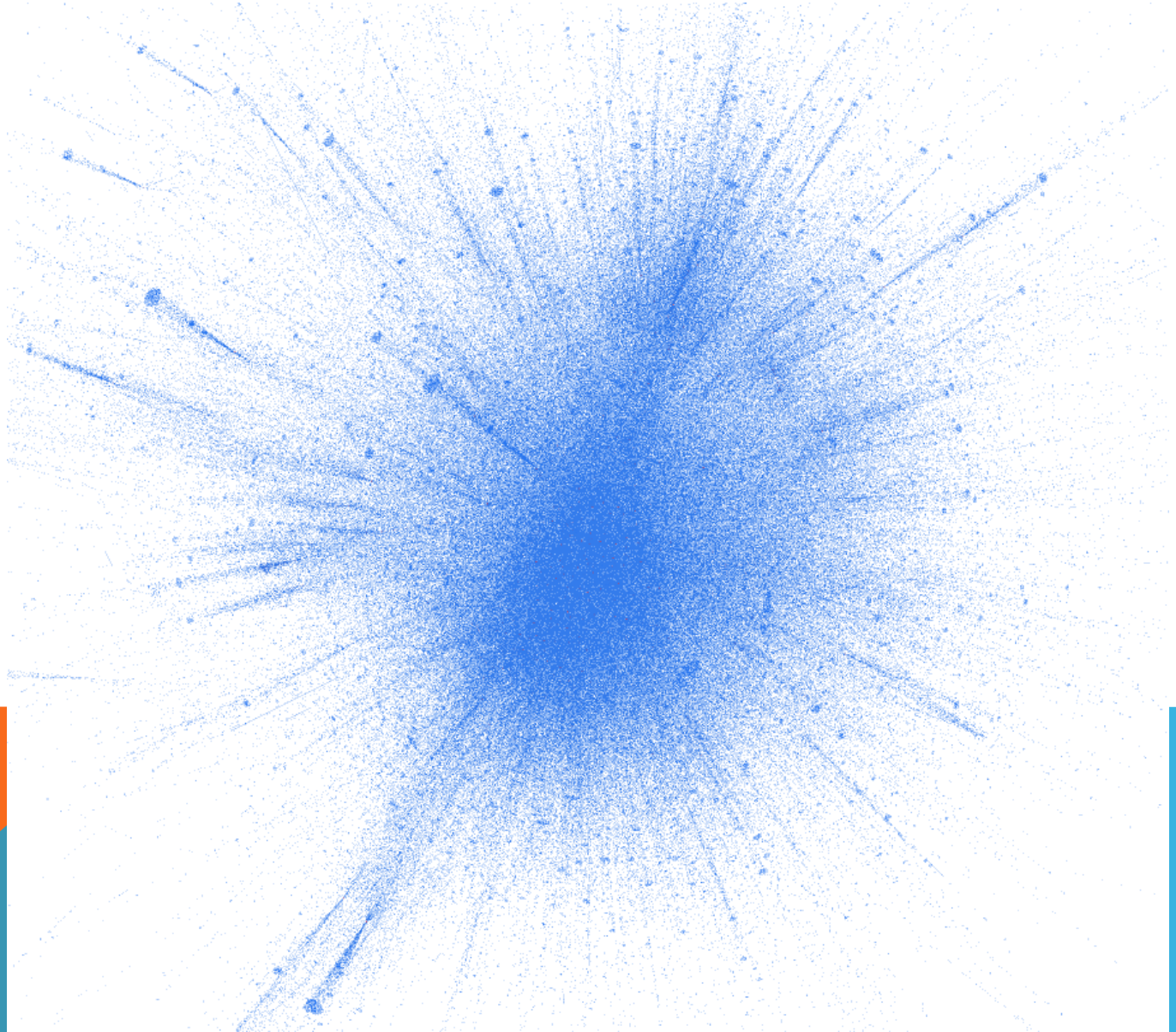
```
Number of NODES in network = 35776
Number of EDGES in network = 124330

The network IS NOT clusterable
------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
7956 EDGES NEED TO BE DELETED
6.4% OF ALL EDGES NEED TO BE DELETED
IN ORDER FOR THE NETWORK TO BECOME CLUSTERABLE, THESE EDGES NEED TO BE DELETED =

-[femranarchy,femradebates]
-[againstmensrights,amrsucks]
-[collapse,adviceanimals]
-[debatealtright,news]
-[blackout2015,announcements]
-[socialistra,srssucks]
-[askstatistics,news]
-[redpillwomen,pics]
-[awwwtf,wtf]
-[littlemissaphrodite,mouthsgonemild]
-[hiphopcirclejerk,hiphopheads]
-[shitamericanssay,chemistry]
-[subredditdrama,drugs]
-[circlebroke,steam]
-[complaints,askreddit]
-[reportthespam,technology]
-[angry,showerthoughts]
-[cryptocurrency,girlgamers]
-[rant,reversegif]
-[recycling,mhocpress]
-[drawforme,tifu]
-[subreddit_stats,impeach_trump]
-[karmacourt,quotesporn]
-[blackladies,politics]
-[cleganebowl,todayilearned]
-[australia,nrl]
-[circlebroke,idubbbz]
-[dentistry,whatisthisthing]
-[mildredditdrama,badlinguistics]
-[circlebroke,unresolvedmysteries]
-[askdocs,asthma]
-[suicidewatch,askwomenadvice]
-[redditinsider,news]
-[britishsuccess,videos]
```

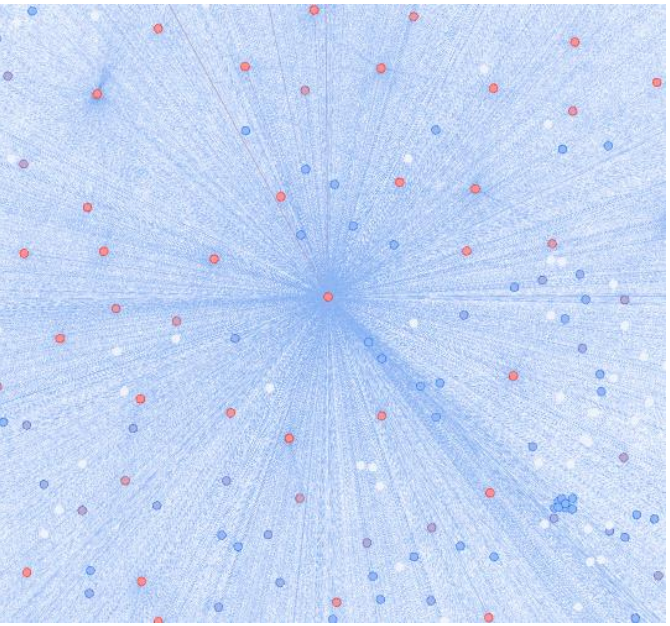
- ЈЕДАН ВЕЛИКИ КЛАСТЕР КОЈИ НИЈЕ КОАЛИЦИЈА
- ГОМИЛА МАЊИХ КЛАСТЕРА КОЈИ ЈЕСУ КОАЛИЦИЈА

```
[onionlovers, onionpeace] - (2 vertices)
[prescott_papers, academicwriting] - (2 vertices)
[that_poppy, theargoproject, marsargo, that_poppy_uncensored] - (4 vertices)
[woot, mehdotcom] - (2 vertices)
[aplang2014, sisypheanhigh] - (2 vertices)
[cracktroubleshoot, madmaxcrack] - (2 vertices)
[ender, endersgame] - (2 vertices)
[scienceillustration, medicalillustration] - (2 vertices)
[vertixclans, vertixonline] - (2 vertices)
[iamtallertssecrets, pmcrebellion] - (2 vertices)
[cryo_kids, donorconceived] - (2 vertices)
[solving_fullemptiness, fullemptiness] - (2 vertices)
[sporecontest, spore] - (2 vertices)
[kent, akron] - (2 vertices)
[ladbanter, gonegonegonegonetron3] - (2 vertices)
NUMBER OF CLUSTERS THAT ARE COALITIONS = 492

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
PROGRAM TOOK = 423.0 SECONDS
```

```
halloweenhelp, orgasm_gif_squirt, 3dsgamenight, postpreview, dhammawheel, westofnuket
infanticide, bikebuilders, custeam, osmproject, skyrimmods_test, glasgow, nosillysuff
tampabayrowdies, behrend, shaders, podemosgijonxixon, gigabyte, dangercart, decodeus,
pogobots, cincinnati, devo, amarisuhc, thetotallyradshow, flexiblesmut, gcdebatesqt,
pedobaittime, aurora, iphonehelp, kamilny, lggwatchr, atlantabarters, festivalsluts,
sixwordplots, dogecoingifs, subredditdramax3, femininеboys, botany, cyberonline, flee
missourigamers, doesthisexist, jinxu27subreddit, bulges, crookedcocks, storj, ctrlalt
navia, hoodedeyes, thelunarsanctum, copypastaph, auto_repair, blokdust, srgg, marketa
3fvape, stevenuniverse, toyexchange, terrehaute, terramaris, oneplusx, pokemon_go_new
communitylocke2017, showing_off_her_ass, mmofps, etheroll, limitless, xxxcitedbrunett
santosfc, djimavic, veronikasgunpla] - (33814 vertices)
NUMBER OF CLUSTERS THAT ARE NOT COALITIONS = 1----------------------------------
```

ВИЗУАЛИЗАЦИЈА МРЕЖЕ





НЕКИ ОД ХАБОВА У МРЕЖИ