

The summer internship at the faculty of computer science

DSBA 18/19



Krauze Natalia, Smotrova Christina, Pavleeva Maria



HIGHER SCHOOL OF ECONOMICS
N A T I O N A L R E S E A R C H U N I V E R S I T Y

Task:

1. For our vk community get data about members and find SCC, dependences to detect subgroups using VK API
2. Connect it and represent as a graph
3. Explore Karger's algorithm and others to find sub - communities

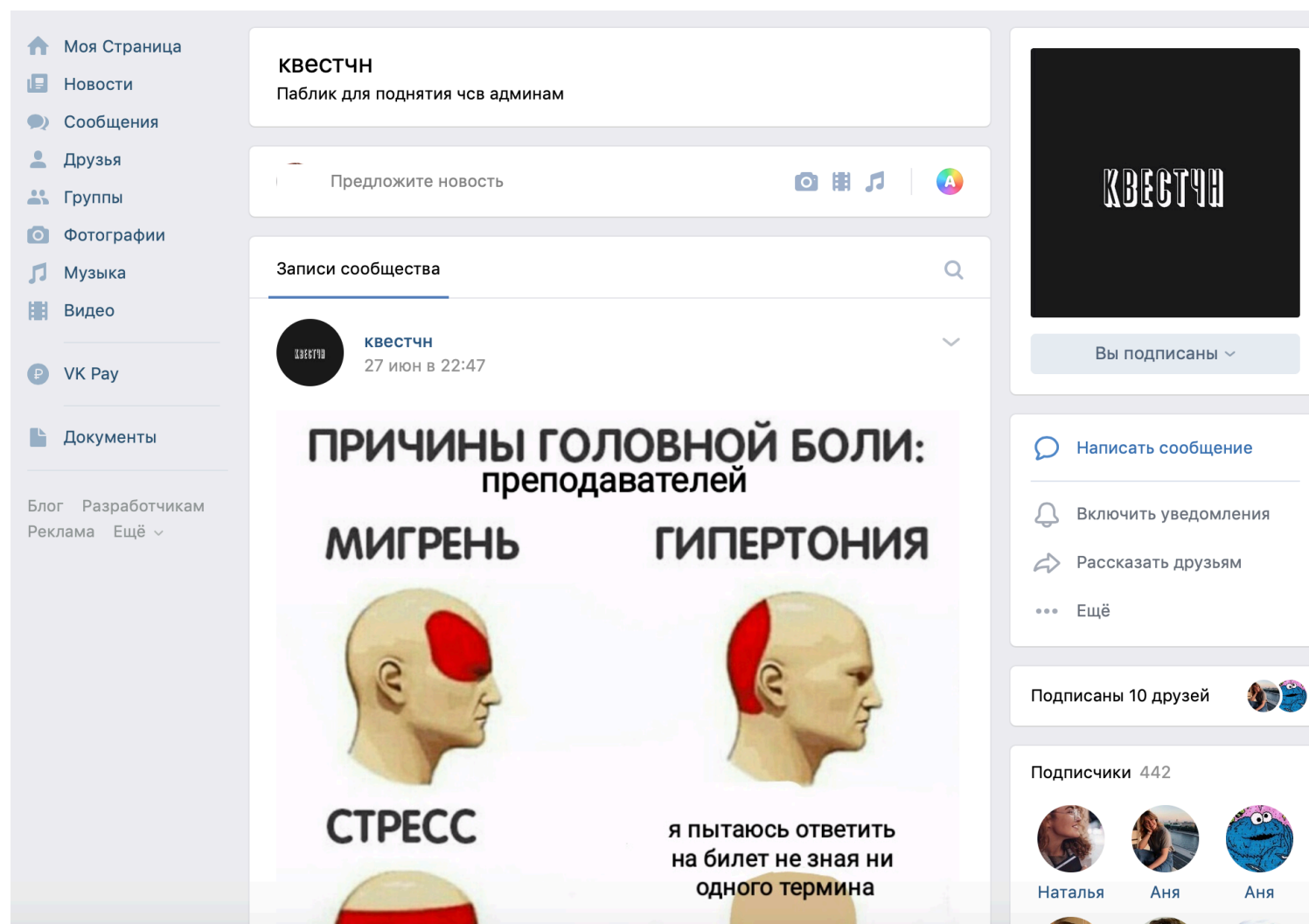


Topic: Community Detection in VK

Object: <https://vk.com/hueschan> (VK group of HSE FCS)

members - 442

chose this group, since the optimal number of subscribers and their relationships with each other - one faculty



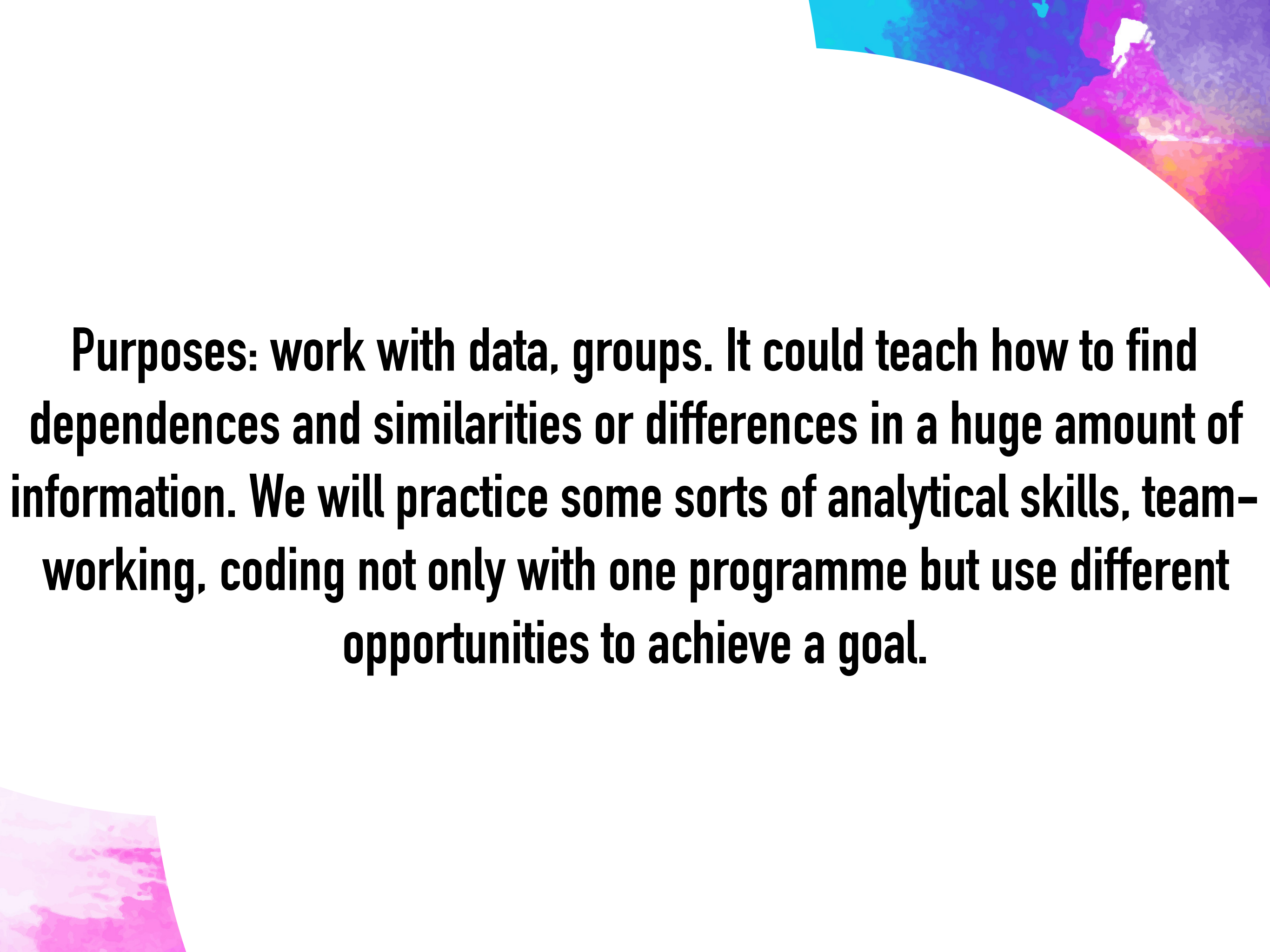
Instruments:

Python 3 (We decided to use Python as it more flexible to integrate with VK API and realising graphs) 

VK API objects

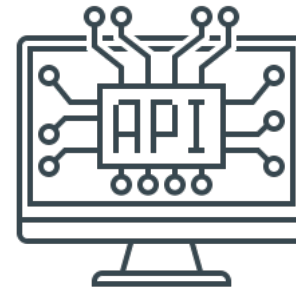
Libraries 'requests', 'NetworkX'

JSON for downloading and processing data



Purposes: work with data, groups. It could teach how to find dependences and similarities or differences in a huge amount of information. We will practice some sorts of analytical skills, team-working, coding not only with one programme but use different opportunities to achieve a goal.

1. GET DATA.VK API



1 – Firstly, we download data about all users of our group.

2 – Then, for everyone if it is possible get a list of his friends.

2. CONSTRUCT A GRAPH

Created an undirected graph, every edge connects users of the group who are friends with each other

The graph is sparse - 2337 edges out of 97682 possible

Why our graph is sparse?

What could be the reason?

1) Some users have closed their profiles, so it was impossible to download a list of their friends. Also some users were banned or have deleted their accounts.

2) Users' conservatism: they don't add to friends all who they know are bypassed

3. Karger's Algorithm



Karger's algorithm is a randomized algorithm to compute a minimum cut of a connected graph. The fundamental operation of Karger's algorithm is a form of edge contraction. the algorithm iteratively contracts randomly chosen edges until only two nodes remain; those nodes represent a cut in the original graph. By iterating this basic algorithm a sufficient number of times, a minimum cut can be found with high probability.

We learned that MIN CUT = MAX FLOW

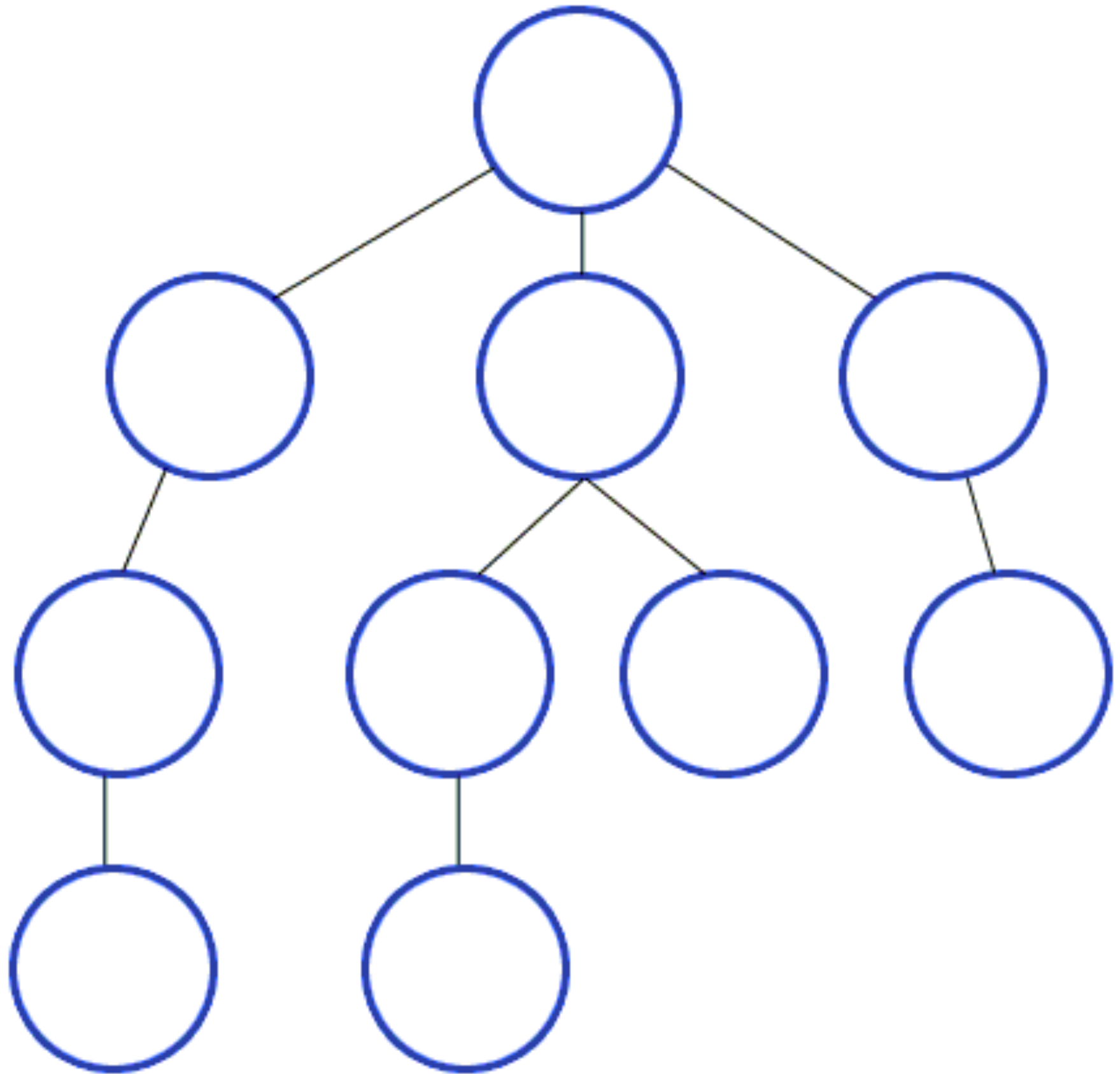
4. Problems with Karger's algorithm

Karger's algorithm works only with connected graphs. In our case, our graph is not connected, the algorithm is not adapted for disconnected graphs, so it cannot be used for the primary partition

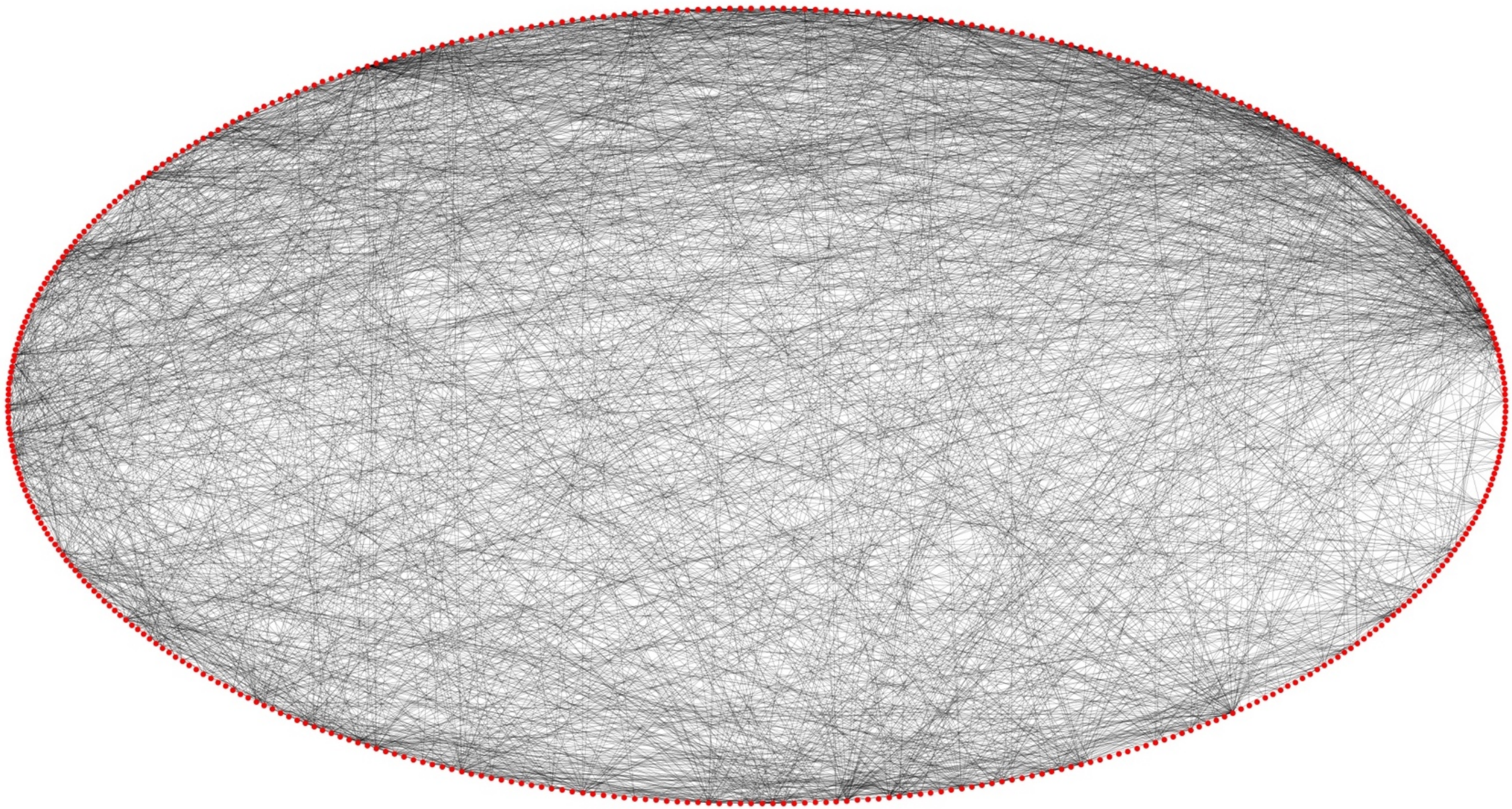
That is why the solution is to find SCC by using DFS algorithm

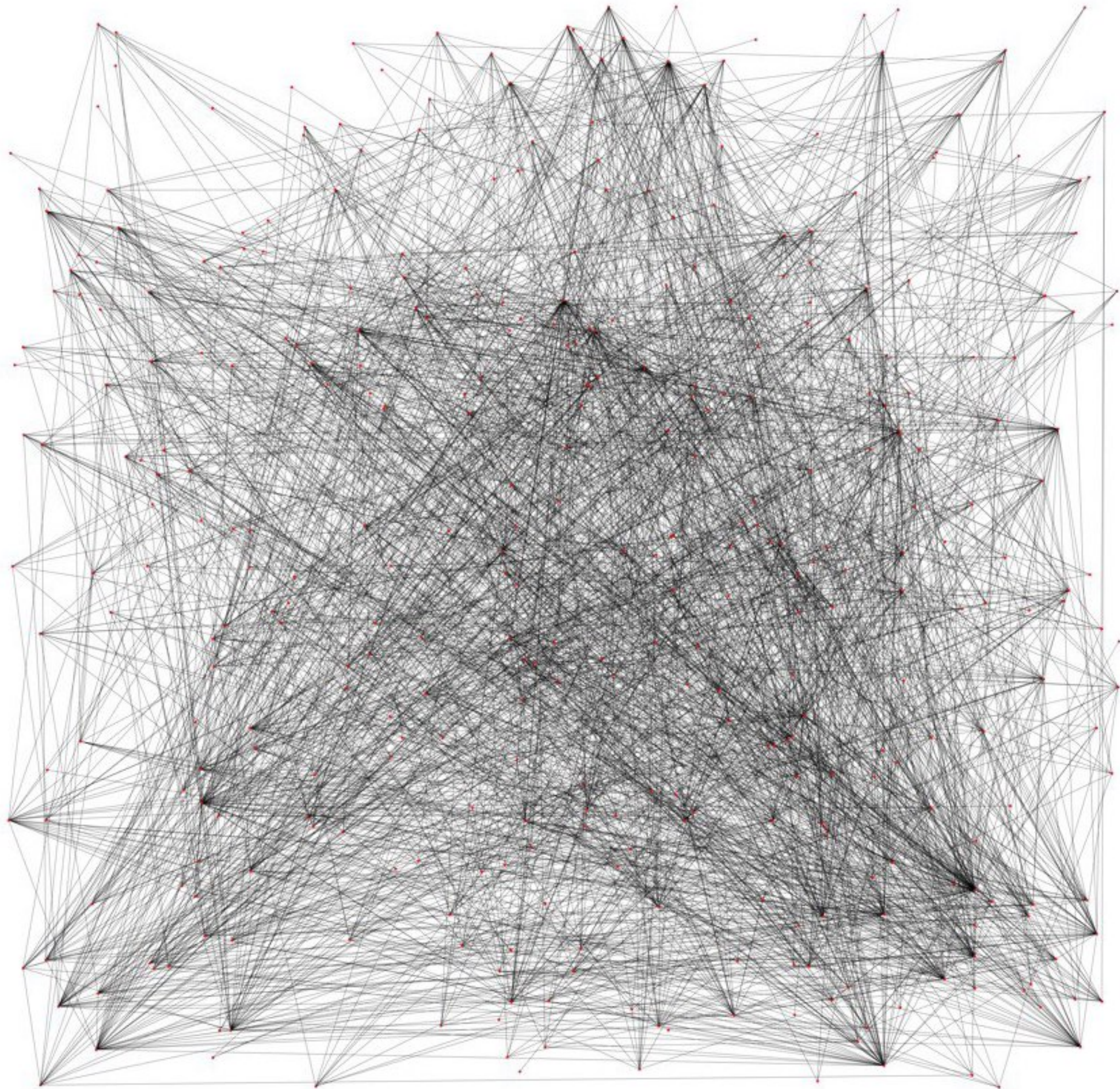
DFS

:



Visualisation of our graph (by using NetworkX)





5. Analysing of SCC

We got a lot of connected components. In our case, one was quite big (about 300 users), others were isolated or pairs of members of our group (for example, pairs of people from other faculties of universities who are interested in topics that the group covers)

Conclusions:

This practice gives us many useful opportunities and skills like: we knew about probability algorithms like the Karger's algorithm. Also, we learned how to work with special apps and libraries in Python as VK API, Request and others. Learned more about theory of graphs, SCC, minimum cut, flows and others. Practice in Python. Worked with visualisations of graphs. Developed team-working skills. In general, we were peached how to work with some kind of analytical problems dedicated to analysing social networks and how to implement and use different algorithm to realise some practical ideas and tasks.