National Research University Higher School of Economics (Higher School of Economics/HSE)
Faculty of Computer Science
Bachelor's Programme Data Science and Business Analytics
01.03.02 Applied Mathematics and Computer Science

Internship report

Fulfilled by

______Smotrova Christina______________________
(Surname, Given Name, Middle Name if any)

____
(signature)

Checked by

________________________________ ____________________ ________________
(job or academic title) (*surname, initials)* (signature)

Moscow, 2019

Internship goals: consolidation, expansion and deepening of the theoretical knowledge and acquirement of initial practical skills in solving specific problems.

Tasks:

1. consolidation and deepening of the theoretical knowledge on the disciplines passed at the university;
2. acquirement of information competence for the purpose of successful work in professional activities;
3. obtaining skills of both independent and team work.

Contents (subjects to be researched, *individual for each student;* you can look it up in the attached file): My responsibilities included writing a Karger's algorithm that works with graphs and testing the code on specific examples. Then we compared our codes with Maria Pavleeva. We chose a group «Квесчн» in VK to be researched.

4. Educational Internship Schedule (Plan)

№	**Calendar period**	**Work Plan**	Internship Supervisor's signature/
1	01.07.2019	1. Organizational (induction) meeting	
2	01.07.2019	2. Instructing on the requirements of labor protection, safety, fire safety and internal labor regulations	
3	01.07.2019 - 13.07.2019	3. Fulfillment of Individual Assignment	
4	01.07.2019 - 13.07.2019	4. Consultation	
5	14.07.2019	5. Preparation and submission of the Report	

5. Description of the learned materials.

I chose Python language to write the Karger's algorithm.

When I started to write an algorithm, I understood that I don't know enough theory about graphs, so I used some books on Discrete Math to help me with that. I faced many problems while implementing my code: sets with nodes and edges were not filled correctly + at the moment where I needed to connect to vertices together, my program worked incorrectly.

6. Description of the results.

I understood how Karger's algorithm works with the help of Maria Pavleeva. We implemented the algorithm for an abstract graph which was connected. But our graph from the data of «Квесчн» was not connected, so it was impossible to use our algorithm on it.

7. Conclusion.

During the summer practice I have learned some new algorithms connected with graphs. Also I revised some materials about graphs in Discrete Math.

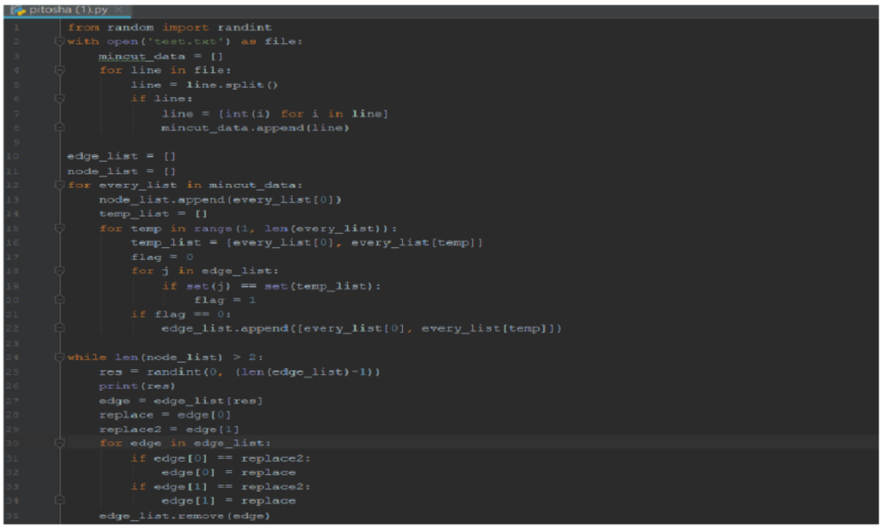
8. Bibliography

https://en.wikipedia.org/wiki/Karger%27s_algorithm https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9A%D0%B0%D1%80%D0%B3%D0%B5%D1%80%D0%B0_%D0%B4%D0%BB%D1%8F_%D0%BD%D0%B0%D1%85%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BC%D0%B8%D0%BD%D0%B8%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D0%B0%D0%B7%D1%80%D0%B5%D0%B7%D0%B0

S. Sydoplatov “Discrete mathematics”

E. Zaripova “Graph theory”

9. Appendix and supplementary material.



```
from random import randint
with open('test.txt') as file:
    mincut_data = []
    for line in file:
        line = line.split()
        if line:
            line = [int(i) for i in line]
            mincut_data.append(line)

edge_list = []
node_list = []
for every_list in mincut_data:
    node_list.append(every_list[0])
    temp_list = []
    for temp in range(1, len(every_list)):
        temp_list = [every_list[0], every_list[temp]]
        flag = 0
        for j in edge_list:
            if set(j) == set(temp_list):
                flag = 1
        if flag == 0:
            edge_list.append([every_list[0], every_list[temp]])

while len(node_list) > 2:
    res = randint(0, (len(edge_list)-1))
    print(res)
    edge = edge_list[res]
    replace = edge[0]
    replace2 = edge[1]
    for edge in edge_list:
        if edge[0] == replace2:
            edge[0] = replace
        if edge[1] == replace2:
            edge[1] = replace
    edge_list.remove(edge)
```