

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

МЕГАФАКУЛЬТЕТ ТРАНСЛЯЦИОННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

ЛАБОРАТОРНАЯ РАБОТА №5

По дисциплине «Введение в цифровую культуру и программирование»

Работа с графом

Выполнил *Шеин Максим Андреевич*

(Фамилия Имя Отчество)



Проверила *Страдина Марина Владимировна*

(Фамилия Имя Отчество)

Санкт-Петербург, 2020г.

Оглавление

Код программы	2
Ход работы	3
Ответы на вопросы	4

Код программы

```
import networkx as f

def minrast(a, b):
    global g
    range = f.single_source_dijkstra(g, a, b)
    return (range)

def step(a):
    global array1
    global array2
    count = 0
    for i in array1:
        if i == a:
            count += 1
    for j in array2:
        if j == a:
            count += 1
    return count

fin = open("graph.txt", "r")

array1 = []
array2 = []
a = []
insulator = []

for i in range(0, 2408):
    i = list(map(int, fin.readline().split()))
    array2.append(i)

delete = [130, 967, 680, 472, 764]

for i in range(len(array2) - 1):
    if not((array2[i][0] % 17 == 0 or array2[i][0] in delete) or
(array2[i][1] % 17 == 0 or array2[i][1] in delete)):
        a.append(array2[i])

print("kolichestvo riober :", len(a))

array1 = []
array2 = []

for i in range(len(a) - 1):
    array1.append(a[i][0])
```

```

        array2.append(a[i][1])

array3 = list(set(array2 + array1))
array4 = [i for i in range(1000)]

for i in array4:
    if not(i in array3 or i in delete or i % 17 == 0):
        insulator.append(i)

print("izolatori :", insulator)

stepversh = []
stepvershfout = []
max1 = 0
max2 = 0
for i in array3:
    degree = step(i)
    stepversh.append([i, degree])
    if degree > max1:
        max1 = degree
        max2 = i

stepvershfout.append([max2, max1])

for i in range(len(stepversh) - 1):
    if stepversh[i][1] == max1 and stepversh[i][0] != max2:
        stepvershfout.append(stepversh[i])

print("vershini s max stepeniu :", stepvershfout)

g = f.MultiGraph()
for i in a:
    g.add_edge(i[0], i[1], range=1)
g = g.to_undirected()

print("diam :", f.diameter(g))
print("rast 734-837 :", minrast(734, 837))
print("rast 608-188:", minrast(608, 188))
print("rast 810-35:", minrast(810, 35))

```

Ход работы

- 1) Создаём массивы под нужды программы.
- 2) Заношу данные из файла в один из массивов.
- 3) Заношу данные о числах, которые надо будет удалить.
- 4) Подсчитываю количество рёбер без вершин, которые кратны 17 и тех, которые мне предоставили, как исключения.
- 5) Создаю массив изоляторов и нахожу их.
- 6) Создаю массив для вершины и её степени.
- 7) Создаю функцию для поиска степени вершины и заполняю массив.
- 8) С помощью networkx создаю граф и нахожу его диаметр.
- 9) Создаю функцию для подсчёта минимального расстояния.
- 10) Вывожу все полученные данные.

Ответы на вопросы

Вопрос №1

Кол-во рёбер - 2407

Вопрос №2

Кол-во изоляторов - 8

Это 32, 222, 278, 425, 668, 688, 735, 997

Вопрос №3

Вершина с самой большой степенью. Это 764 13

Вопрос №4

Диаметр - 9

Вопрос №5

Кратчайший путь(734-837) - это 2. (734, 969, 837)

Вопрос №6

Кратчайший путь(608-188) - это 6 (608, 953, 386, 680, 450, 472, 188)

Вопрос №7

Кратчайший путь(810-35) - это 3 (810, 155, 335, 35)

Вопрос №8

Кол-во рёбер - 2120

Вопрос №9

Кол-во изоляторов - 9

Это 32, 222, 278, 668, 688, 695, 735, 886, 997

Вопрос №10

Вершины с самой большой степенью.

Это 64 12 и 543 12

Вопрос №11

Диаметр - 9

Вопрос №12

Кратчайший путь(734-837) - это 5 (734, 656, 447, 812, 689, 837)

Вопрос №13

Кратчайший путь(608-188) - это 6 (608, 509, 501, 823, 486, 24, 188)

Вопрос №14

Кратчайший путь(810-35) - это 3 (810, 155, 335, 35)