

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский
университет
информационных технологий, механики и оптики»

Факультет информационных технологий и программирования

Кафедра информационных систем

Лабораторная работа № 5

Выполнил студент группы № М3110
Дуксин Патрик

САНКТ-ПЕТЕРБУРГ
2020
Работа с графом

Работа выполнена на языке Python3 с использованием библиотеки networkx.

Ход работы:

Сделан ввод из файла, в массив edges записаны пары рёбер.

```
import networkx as nx

file = open('graphedges133.txt', 'r')
edge = file.readlines()
file.close()
edges = [[i[0], i[1]] for i in ([i.split() for i in edge])]
```

Создан граф graph, добавлены вершины и ребра.

```
graph = nx.Graph()
graph.add_nodes_from([str(i) for i in range(1000)])
graph.add_edges_from(edges)
```

Посчитаны степени вершин и записаны в список degree_sequence.

```
degree_sequence = sorted([n, d] for n, d in graph.degree()), key=lambda x: -x[1])
```

Найдена компонента связности и записана в отдельный граф component.

```
component = nx.Graph()
component.add_nodes_from(list(list(nx.connected_components(graph))[0]))
component.add_edges_from(edges)
```

Посчитаны кратчайшие пути между всеми вершинами графа, записаны в словарь distance.

```
distance = dict(nx.all_pairs_shortest_path(graph))
```

Создана копия графа graph удалены необходимые элементы, получившийся граф записан в new_graph.

```
new_graph = graph.copy()
new_graph.remove_nodes_from([str(i) for i in range(0, 1000, 17)])
new_graph.remove_nodes_from(list(map(str, [480, 771, 595, 58, 794, 667, 922])))
```

Далее для расчетов применена аналогичная логика, что и для graph.

```
new_degree_sequence = sorted([n, d] for n, d in new_graph.degree()), key=lambda x: -x[1])

new_component = nx.Graph()
new_component.add_nodes_from(list(list(nx.connected_components(new_graph))[0]))
new_component.add_edges_from(edges)

new_distance = dict(nx.all_pairs_shortest_path(new_graph))
```

Вывод ответов на вопросы:

```
out = open('output.txt', 'w')
out.write(f'Вопрос 1:\n\t'
        f'Число ребёр в графе: {nx.number_of_edges(graph)}\n')
out.write(f'Вопрос 2:\n\t'
        f'Число изолятов в графе: {nx.number_of_isolates(graph)}\n\t'
        f'Изоляты: {" ".join(sorted(list(nx.isolates(graph))))}\n')
out.write(f'Вопрос 3:\n\t'
        f'Вершина с самой большой степенью: {degree_sequence[0][0]}, степень: {degree_sequence[0][1]},\n\t'
        f'Вершины связанные с {degree_sequence[0][0]}: '
        f'{" ".join([i[1] for i in list(graph.edges(degree_sequence[0][0]))])}\n')
out.write(f'Вопрос 4:\n\t'
        f'Диаметр компоненты связности: {nx.diameter(component)}\n')
out.write(f'Вопрос 5:\n\t'
        f'Расстояние между 812 и 790: '
        f'{len(distance["812"]["790"])}. Путь: {"->".join(distance["812"]["790"])}\n')
out.write(f'Вопрос 6:\n\t'
        f'Расстояние между 859 и 745: '
        f'{len(distance["859"]["745"])}. Путь: {"->".join(distance["859"]["745"])}\n')
out.write(f'Вопрос 7:\n\t'
        f'Расстояние между 266 и 278: '
        f'{len(distance["266"]["278"])}. Путь: {"->".join(distance["266"]["278"])}\n')
out.write(f'Вопрос 8:\n\t'
        f'Число ребёр в графе: {nx.number_of_edges(new_graph)}\n')
```

```

out.write(f'Вопрос 8:\n\t'
        f'Число ребёр в графе: {nx.number_of_edges(new_graph)}\n')
out.write(f'Вопрос 9:\n\t'
        f'Число изолятов в графе: {nx.number_of_isolates(new_graph)}\n\t'
        f'Изоляты: {" ".join(sorted(list(nx.isolates(new_graph))))}\n')
out.write(f'Вопрос 10:\n\t'
        f'Вершина с самой большой степенью: {new_degree_sequence[0][0]}, степень: {new_degree_sequence[0][1]}\n\t'
        f'Вершины связанные с {new_degree_sequence[0][0]}: '
        f'{" ".join([i[1] for i in list(graph.edges(new_degree_sequence[0][0]))])}\n')
out.write(f'Вопрос 11:\n\t'
        f'Диаметр компоненты связности: {nx.diameter(new_component)}\n')
out.write(f'Вопрос 12:\n\t'
        f'Расстояние между 812 и 790: '
        f'{len(distance["812"]["790"])}. Путь: {"->".join(new_distance["812"]["790"])}\n')
out.write(f'Вопрос 13:\n\t'
        f'Расстояние между 859 и 745: '
        f'{len(distance["859"]["745"])}. Путь: {"->".join(new_distance["859"]["745"])}\n')
out.write(f'Вопрос 14:\n\t'
        f'Расстояние между 266 и 278: '
        f'{len(distance["266"]["278"])}. Путь: {"->".join(new_distance["266"]["278"])}\n')
out.close()

```

Ответы на вопросы:

Вопрос 1:

Число ребёр в графе: 2486

Вопрос 2:

Число изолятов в графе: 7

Изоляты: 255 328 394 395 434 677 724

Вопрос 3:

Вершина с самой большой степенью: 624, степень: 15,

Вершины связанные с 624: 35 156 252 295 312 375 380 467 551 784 803 880 896 922 955

Вопрос 4:

Диаметр компоненты связности: 9

Вопрос 5:

Расстояние между 812 и 790: 7. Путь: 812->311->203->922->783->744->790

Вопрос 6:

Расстояние между 859 и 745: 6. Путь: 859->58->354->794->595->745

Вопрос 7:

Расстояние между 266 и 278: 5. Путь: 266->667->480->771->278

Вопрос 8:

Число ребёр в графе: 2165

Вопрос 9:

Число изолятов в графе: 8

Изоляты: 328 394 395 434 677 713 724 773

Вопрос 10:

Вершина с самой большой степенью: 624, степень: 14,

Вершины связанные с 624: 35 156 252 295 312 375 380 467 551 784 803 880
896 922 955

Вопрос 11:

Диаметр компоненты связности: 9

Вопрос 12:

Расстояние между 812 и 790: 7. Путь: 812->582->548->584->178->744->790

Вопрос 13:

Расстояние между 859 и 745: 6. Путь: 859->202->872->946->921->745

Вопрос 14:

Расстояние между 266 и 278: 5. Путь: 266->899->302->932->278

Код работы:

```
import networkx as nx
```

```
file = open('graphedges133.txt', 'r')  
edge = file.readlines()  
file.close()
```

```
edges = [[i[0], i[1]] for i in ([i.split() for i in edge])]
```

```
graph = nx.Graph()  
graph.add_nodes_from([str(i) for i in range(1000)])  
graph.add_edges_from(edges)
```

```
degree_sequence = sorted([(n, d) for n, d in graph.degree()], key=lambda x: -x[1])
```

```
component = nx.Graph()  
component.add_nodes_from(list(list(nx.connected_components(graph))[0]))  
component.add_edges_from(edges)
```

```
distance = dict(nx.all_pairs_shortest_path(graph))
```

```
new_graph = graph.copy()  
new_graph.remove_nodes_from([str(i) for i in range(0, 1000, 17)])  
new_graph.remove_nodes_from(list(map(str, [480, 771, 595, 58, 794, 667, 922])))
```

```
new_degree_sequence = sorted([(n, d) for n, d in new_graph.degree()], key=lambda x: -x[1])
```

```
new_component = nx.Graph()  
new_component.add_nodes_from(list(list(nx.connected_components(new_graph))[0]))  
new_component.add_edges_from(edges)
```

```
new_distance = dict(nx.all_pairs_shortest_path(new_graph))
```

```
out = open('output.txt', 'w')  
out.write(f'Вопрос 1:\n\t'  
         f'Число ребёр в графе: {nx.number_of_edges(graph)}\n')  
out.write(f'Вопрос 2:\n\t')
```

```

    f'Число изолятов в графе: {nx.number_of_isolates(graph)}\n\t'
    f'Изоляты: {" ".join(sorted(list(nx.isolates(graph))))}\n')
out.write(f'Вопрос 3:\n\t'
    f'Вершина с самой большой степенью: {degree_sequence[0][0]}, степень:
{degree_sequence[0][1]},\n\t'
    f'Вершины связанные с {degree_sequence[0][0]}: '
    f'{" ".join([i[1] for i in list(graph.edges(degree_sequence[0][0]))])}\n')
out.write(f'Вопрос 4:\n\t'
    f'Диаметр компонента связности: {nx.diameter(component)}\n')
out.write(f'Вопрос 5:\n\t'
    f'Расстояние между 812 и 790: '
    f'{len(distance["812"]["790"])}. Путь: {"->".join(distance["812"]["790"])}\n')
out.write(f'Вопрос 6:\n\t'
    f'Расстояние между 859 и 745: '
    f'{len(distance["859"]["745"])}. Путь: {"->".join(distance["859"]["745"])}\n')
out.write(f'Вопрос 7:\n\t'
    f'Расстояние между 266 и 278: '
    f'{len(distance["266"]["278"])}. Путь: {"->".join(distance["266"]["278"])}\n')

out.write(f'Вопрос 8:\n\t'
    f'Число ребёр в графе: {nx.number_of_edges(new_graph)}\n')
out.write(f'Вопрос 9:\n\t'
    f'Число изолятов в графе: {nx.number_of_isolates(new_graph)}\n\t'
    f'Изоляты: {" ".join(sorted(list(nx.isolates(new_graph))))}\n')
out.write(f'Вопрос 10:\n\t'
    f'Вершина с самой большой степенью: {new_degree_sequence[0][0]}, степень:
{new_degree_sequence[0][1]},\n\t'
    f'Вершины связанные с {new_degree_sequence[0][0]}: '
    f'{" ".join([i[1] for i in list(graph.edges(new_degree_sequence[0][0]))])}\n')
out.write(f'Вопрос 11:\n\t'
    f'Диаметр компонента связности: {nx.diameter(new_component)}\n')
out.write(f'Вопрос 12:\n\t'
    f'Расстояние между 812 и 790: '
    f'{len(distance["812"]["790"])}. Путь: {"->".join(new_distance["812"]["790"])}\n')
out.write(f'Вопрос 13:\n\t'
    f'Расстояние между 859 и 745: '
    f'{len(distance["859"]["745"])}. Путь: {"->".join(new_distance["859"]["745"])}\n')
out.write(f'Вопрос 14:\n\t'
    f'Расстояние между 266 и 278: '
    f'{len(distance["266"]["278"])}. Путь: {"->".join(new_distance["266"]["278"])}\n')
out.close()

```