**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Российский химико-технологический университет имени Д.И. Менделеева»**

**Кафедра информационных компьютерных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5**

Выполнил студент группы КС-36 Перминова П.А.

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Дата сдачи:

**Оглавление**

[**Описание задачи**](#_os6af9yzriow) **3**

[**Описание метода/модели**](#_qmtzjq1uu4fo) **4**

[**Выполнение задачи**](#_qei8cgvtd8ep) **5**

[**Заключение**](#_5uowgjmy6b3u) **6**

# Описание задачи

Создайте взвешенный граф, состоящий из [10, 20, 50, 100] вершин.

* Каждая вершина графа связана со случайным количеством вершин, минимум с [3, 4, 10, 20].
* Веса ребер задаются случайным значением от 1 до 20.
* Каждая вершина графа должна быть доступна, т.е. до каждой вершины графа должен обязательно существовать путь до каждой вершины, не обязательно прямой.

Выведите получившийся граф в виде матрицы смежности.

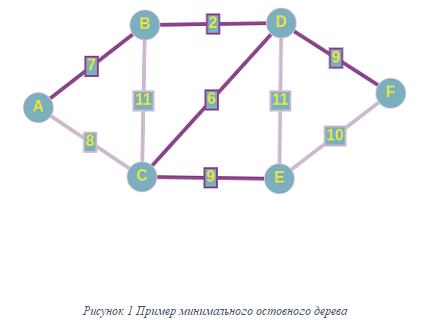
Для каждого графа требуется провести серию из 5-10 тестов, в зависимости от времени затраченного на выполнение одного теста, необходимо построить минимальное остовное дерево взвешенного связного неориентированного графа с помощью алгоритма Прима.

В рамках каждого теста необходимо замерить потребовавшееся время на выполнение задания из пункта 3 для каждого набора вершин. По окончанию всех тестов необходимо построить график используя полученные замеры времени, где на ось абсцисс (Х) нанести N – количество вершин, а на ось ординат(Y) - значения затраченного времени.

# Описание метода/модели

Суть самого алгоритма Прима сводится к жадному перебору рёбер из определенного множества. На входе имеется пустой подграф, который и будем достраивать до потенциального минимального остовного дерева.

* Изначально наш подграф состоит из одной любой вершины исходного графа.
* Затем из рёбер инцидентных этой вершине, выбирается такое минимальное ребро, которое связала бы две абсолютно разные компоненты связности, одной из которых и является наш подграф. То есть, как только у нас появляется возможность добавить новую вершину в наш подграф, мы тут же включаем ее по минимально возможному весу.
* Продолжаем выполнять предыдущий шаг до тех пор, пока не найдем искомое MST.



# Выполнение задачи

Данная лабораторная работа была реализована на языке Python.

Классы, реализованные в программе:

* Generate\_graph
* Graph (из реализованного в данном классе метода create\_adjacency\_matrix() можно получить матрицу смежности; также в prim() реализован алгоритм Прима.)

Данные о зависимости времени выполнения алгоритма Прима от количества вершин графа, полученные в ходе выполнения программы, представлены на графике.



# 

# Заключение

В ходе выполнения лабораторной работы были сделаны следующие выводы о достоинствах и недостатках алгоритма Прима.

Достоинства алгоритма Прима: гарантия получения глобального оптимального решения.

Недостатки: алгоритм применим только к неориентированным графам.