МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА 25

КУРСОВАЯ РАБОТА (ПРОЕКТ)
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

Доцент, канд. техн. наук		Е. М. Линский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

АЛГОРИТМ ПРИМА

по дисциплине: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 2354		К.С. Радионова
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2025

Содержание

1 Постановка задачи

Задачей данной курсовой работы является разработка программы, которая реализует алгоритм Прима. Данный алгоритм ищет минимальное остовное дерево - остовное дерево с минимальным возможным весом. Например, исключив определённые рёбра из графа мы получим подграф, где все вершины по-прежнему связаны друг с другом, но с минимально возможной суммой весов включенных рёбер. Граф должен быть взвешенным и неориентированным.

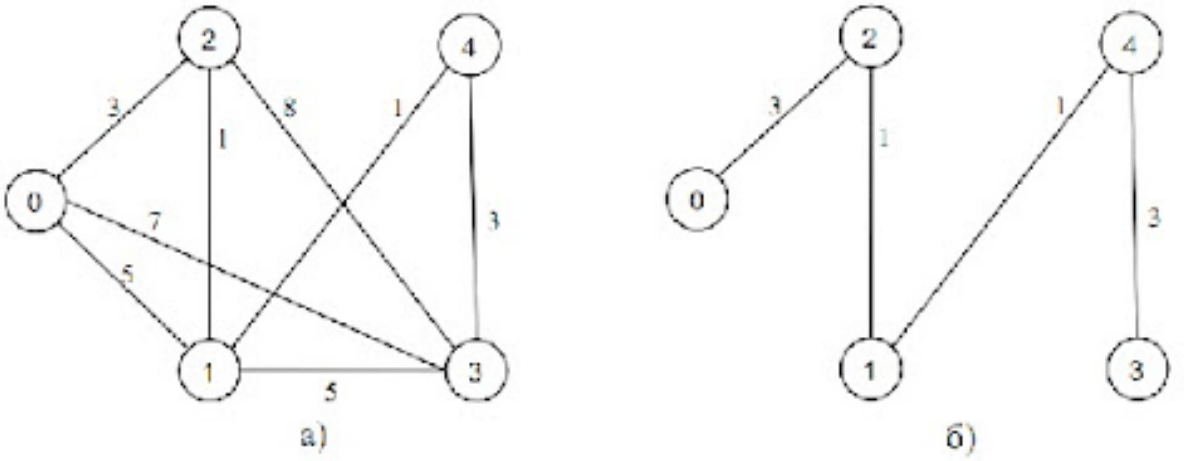


Рисунок 1 - Пример результата работы алгоритма Прима

В книге «Построение и анализ вычислительных алгоритмов» достаточно доступно написано про данный алгоритм (https://www.rulit.me/books/postroenie-i-analiz-vychislitelnyh-algoritmov-download-675866.html).

1.1 Постановка дополнительной задачи

Вводить граф с консоли. Добавить проверку на связность графа при вводе с консоли. Если граф связный, то выполнять алгоритм, если нет – повторно вводить граф.

2 Алгоритм

Суть самого алгоритма Прима сводится к жадному перебору рёбер, но уже из определенного множества. На входе имеется пустой подграф, который будет достраиваться 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 (минимальное остовное дерево).

2.1 Пошаговый алгоритм Прима на примере

1. На вход мы получили пустой подграф с вершинами и известными весами рёбер. Пусть произвольно выбранная вершина – это вершина A.

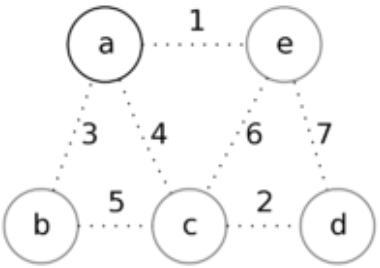


Рисунок 2 - Шаг 1

2. Извлечём из множества вершину a, так как её приоритет минимален. Рассмотрим смежные с ней вершины b, c, и e. Обновим их приоритеты, как веса соответствующих рёбер ab, ac и ae, которые будут добавлены в ответ.

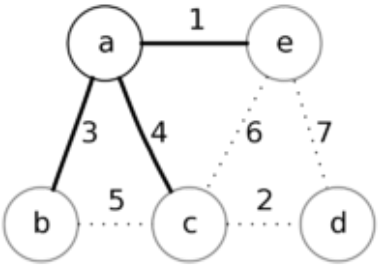


Рисунок 3 - Шаг 2

3. Теперь минимальный приоритет у вершины e. Извлечём её и рассмотрим смежные с ней вершины a, c, и d. Изменим приоритет только у вершины d, так как приоритеты вершин a и c меньше, чем веса у соответствующих рёбер ea и ec, и установим приоритет вершины d равный весу ребра ed, которое будет добавлено в ответ.

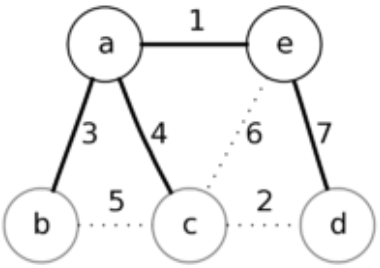


Рисунок 4 - Шаг 3

4. После извлечения вершины b ничего не изменится, так как приоритеты вершин a и c меньше, чем веса у соответствующих рёбер ba и bc. Однако, после извлечения следующей вершины — c, будет обновлён приоритет у вершины d на более низкий (равный весу ребра cd) и в ответе ребро ed будет заменено на cd.

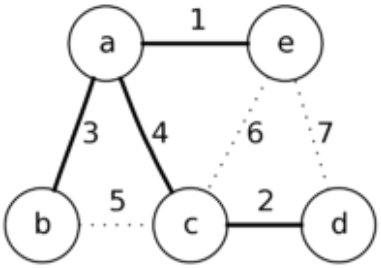


Рисунок 5 - Шаг 4

Далее будет рассмотрена следующая вершина — d, но ничего не изменится, так как приоритеты вершин e и c меньше, чем веса у соответствующих рёбер de и dc. После этого алгоритм завершит работу, так как в заданном множестве не останется вершин, которые не были бы рассмотрены.

2.2 Псевдокод реализации алгоритма

```
Функция PrimMST(G):  // G - взвешенный неориентированный граф
Вход: Граф G с N вершинами
Выход: Список рёбер минимального остовного дерева (MST) и его общий вес
1. Создать пустое множество MST
2. Инициализировать массив visited длины N значением False
3. Создать мин-кучу (приоритетную очередь) edges
4. Выбрать произвольную начальную вершину (например, 0)
5. Добавить все рёбра, инцидентные начальной вершине, в очередь edges
6. Пометить начальную вершину как посещенную

Пока в очереди edges есть элементы:

7. Достать ребро (u, v) с минимальным весом из edges
8. Если вершина v уже посещена, пропустить это ребро
9. Добавить ребро (u, v) в MST
10. Обновить общий вес MST
11. Пометить вершину v как посещенную
12. Добавить все рёбра, инцидентные v и ведущие к непосещённым вершинам, в edges
13. Вернуть MST и его общий вес
```

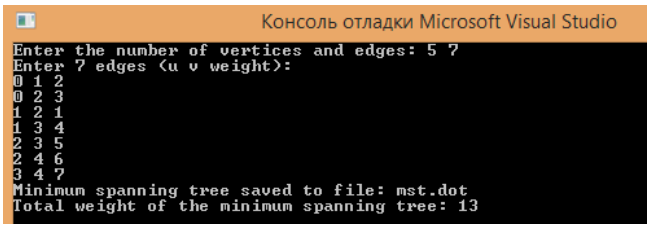
Так как мы используем кучу для хранения ключей, то мы можем извлекать минимальный элемент за O(log V) и обновлять ключи соседних вершин за O(log V) для каждого ребра. В этом случае, если мы обрабатываем все рёбра, общая сложность будет O(E log V), где E – кол-во рёбер, а V – кол-во вершин графа

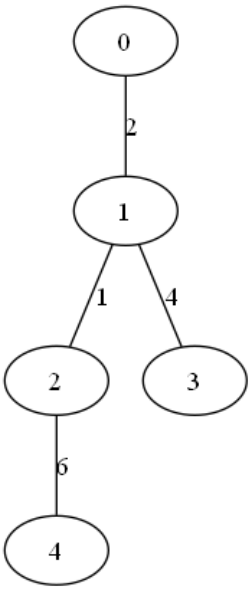
3 Инструкция пользователя

Пользователю необходимо ввести с консоли количество вершин и рёбер графа, а также сам граф в виде списка рёбер. Далее произойдёт проверка на связность графа. Если пользователь ввёл несвязный граф, то ему будет необходимо заново ввести граф, а если связный – программа выполнится. В результате работы программы пользователь получит минимальное остовное дерево. Оно сохранится в виде файла mst.dot, а также в консоль будет выведен общий вес минимального остовного дерева. Чтобы сохранить файл mst.dot в виде файла png, нужно вызвать командную строку и ввести “cd адрес файла”, далее команду “dot -Tpng mst.dot -o mst.png” и в папку с проектом сохранится в папку с проектом.

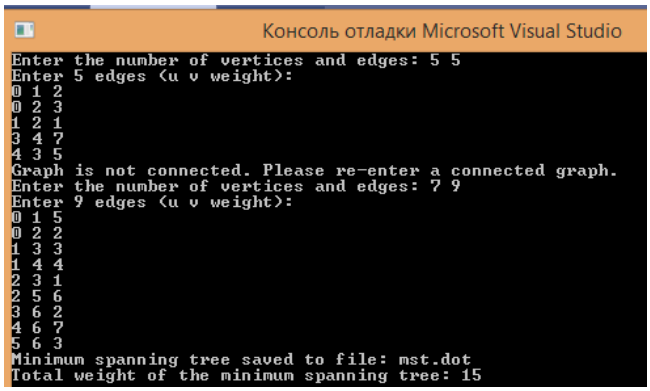
4 Тестовые примеры

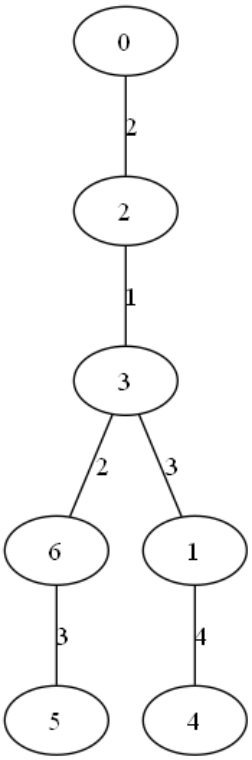
Тест 1:



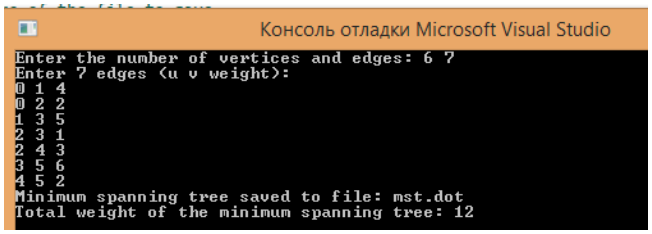


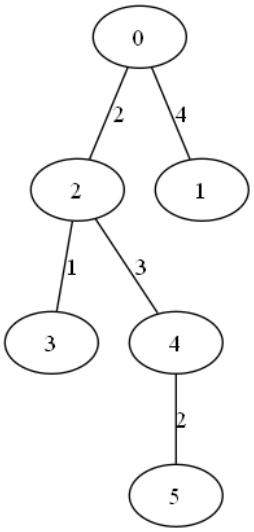
Тест 2:


Консоль отладки Microsoft Visual Studio
Enter the number of vertices and edges: 5 5
Enter 5 edges (u v weight):
0 1 2
0 2 3
1 2 1
3 4 7
4 3 5
Graph is not connected. Please re-enter a connected graph.
Enter the number of vertices and edges: 7 9
Enter 9 edges (u v weight):
0 1 5
0 2 2
1 3 3
1 4 4
2 3 1
2 5 6
3 6 2
4 6 7
5 6 3
Minimum spanning tree saved to file: mst.dot
Total weight of the minimum spanning tree: 15


0
2
2
1
3
2
3
6
1
3
4
5
4

Тест 3:


Консоль отладки Microsoft Visual Studio
Enter the number of vertices and edges: 6 7
Enter 7 edges (u v weight):
0 1 4
0 2 2
1 3 5
2 3 1
2 4 3
3 5 6
4 5 2
Minimum spanning tree saved to file: mst.dot
Total weight of the minimum spanning tree: 12


0
2
4
2
1
1
3
3
4
2
5

5 Список литературы

1. А. Ахо, Дж. Хопкрофт, Дж.Ульман, Построение и анализ вычислительных алгоритмов, Мир, 1979г.
2. Алгоритм Краскала, Прима для нахождения минимального остовного дерева, 2021г. URL: https://habr.com/ru/articles/569444/
3. Минимальное остовное дерево. Алгоритм Прима. Алгоритм Крускала, 2023г. URL: https://brestprog.by/topics/mst/