# **Формулировка задания:**

Реализовать визуализацию алгоритма построения минимального остовного дерева заданного графа(ЯПД) на языке Java.

# **Спецификация:**

# **Описание алгоритма:**

На вход алгоритма подаётся связный неориентированный граф. Для каждого ребра задаётся его стоимость.

Сначала берётся произвольная вершина и находится ребро, инцидентное данной вершине и обладающее наименьшей стоимостью. Найденное ребро и соединяемые им две вершины образуют дерево. Затем, рассматриваются рёбра графа, один конец которых — уже принадлежащая дереву вершина, а другой — нет; из этих рёбер выбирается ребро наименьшей стоимости. Выбираемое на каждом шаге ребро присоединяется к дереву. Таким образом, при выполнении каждого шага алгоритма, высота формируемого дерева увеличивается на 1. Рост дерева происходит до тех пор, пока не будут исчерпаны все вершины исходного графа.

Результатом работы алгоритма является остовное дерево минимальной стоимости.

# **Псевдокод:**

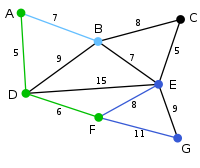
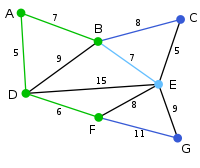
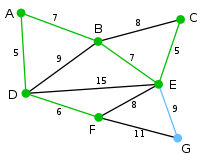
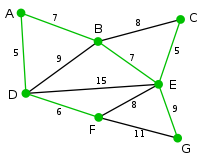
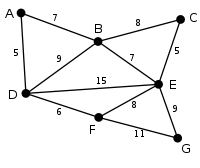
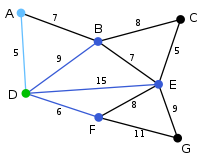
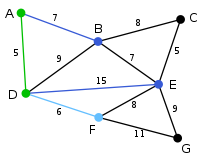
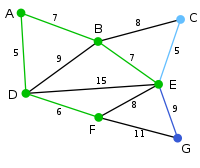
Обозначения:

# **Оценка:**

Асимптотика алгоритма зависит от способа хранения графа и способа хранения вершин, не входящих в дерево. Если приоритетная очередь Q реализована как обычный массив d, то Extract.min(Q) выполняется за O(n), а стоимость операции составляет O(1). Если Q представляет собой бинарную пирамиду, то стоимость Extract.min(Q) снижается до При использовании пирамид Фибоначчи, операция Extract.min(Q) выполняется за , а за O (1).

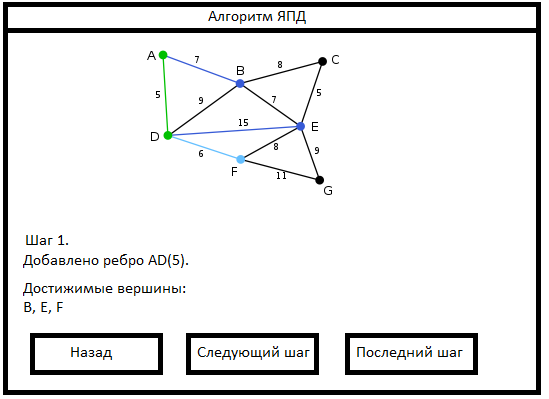
Выбранный способ хранения – массив матрицы смежности.

# **Пример выполнения:**



Зеленым отмечены вершины и ребра, включенные в минимальное остовное дерево, синим – вершины и ребра, доступные для включения на текущем шаге, голубым – наиболее оптимальное ребро на текущем шаге для добавления в минимальное остовное дерево.

# **Интерфейс программы(набросок):**



Кнопка «Назад» ведет к диалоговому окну ввода параметров графа.

Возможные варианты ввода:

* Подключаемый текстовый файл(.txt);
* Ввод в поле диалогового окна (поле текстового ввода, аналог Text box);
* Ввод в поле диалогового окна и из текстового файла производится в следующем формате:
* 1-я строка – 2 целых числа, обозначающие число вершин и число ребер графа соответственно;
* Последующие строки: по 3 числа, обозначающие номера двух вершин, соединяемых ребром, и длину ребра соответственно.
* Ручное задание графа в графическом виде при помощи мыши (графическое поле с осями координат, двойной клик в поле для отметки вершины графа, соединить две вершины с зажатой левой кнопкой мыши для создания ребра между ними)

Кнопка «Следующий шаг» вызывает однократное выполнение тела цикла алгоритма. Кнопка остается активной до полного выполнения алгоритма.

Кнопка «Последний шаг» вызывает выполнение алгоритма полностью с отрисовкой конечного остовного дерева.

В поле «Шаг» отображается номер выполняемого шага алгоритма (без учета выбора начальной вершины), в поле «Достижимые вершины» отображаются вершины графа, для которых выполняется алгоритм на текущем шаге.

Распределение ролей в бригаде:

Андросов: реализация графического интерфейса для демонстрации графа и минимального остовного дерева.

Исмаилов: реализация инструментов демонстрации графов.

Степанов: реализация выполняемого алгоритма(ЯПД).

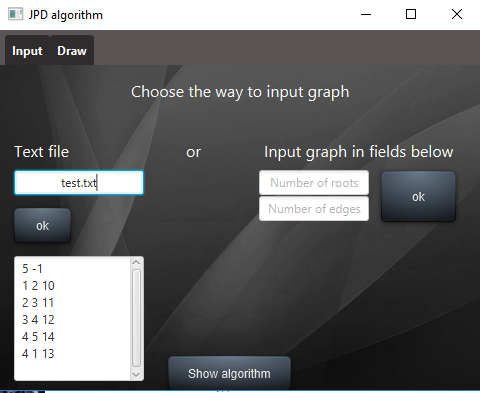
# **План разработки:**

|  |  |
| --- | --- |
| 27.06 | Итоговое согласование спецификации, создание прототипа GUI, разработка плана тестирования |
| 28.06 | Реализация рабочего алгоритма ЯПД |
| 28.06-29.06 | Реализация 1-й версии программы, тестирование, разработка документации. В 1-й версии программы необходимо реализовать ввод из файла, либо из поля диалогового окна (достаточно один из двух), а также отрисовку введенного графа и последнего шага алгоритма (готовое остовное дерево). |
| 30.06-01.07 | Реализация 2-й версии программы, тестирование. Во 2-й версии необходимо реализовать ввод из файла и из файла диалогового окна (оба варианта), а также отрисовку пошагового выполнения алгоритма. |
| 03.07 | Реализация финальной версии программы, сдача программы и документации. В финальной версии необходимо реализовать ввод графа в графическом поле с помощью мыши. |

# **Тестирование**

Первая версия:

Содержимое файла text.txt:



5 -1

1 2 10

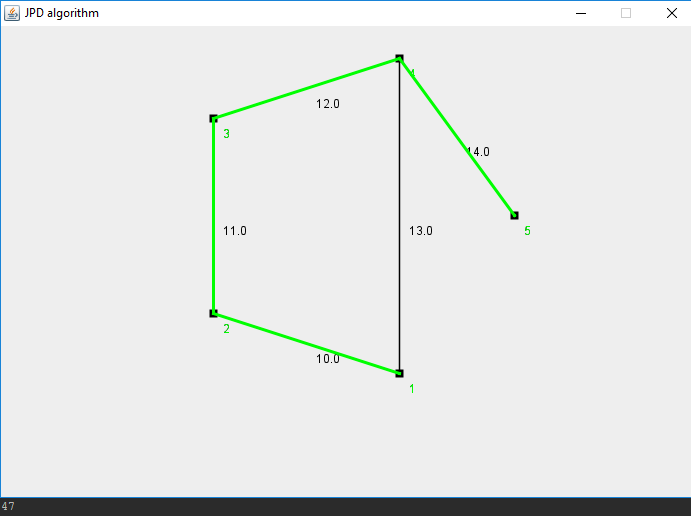
2 3 11

3 4 12

4 5 14

4 1 13

Граф и минимальное остовное дерево, построенные на данных из файла test.txt:



При некорректном вводе файла:

