САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

*Кафедра информатики и прикладной математики*

**Лабораторная работа №3**

*«Нахождение минимального остова графа»*

Выполнил:

студент II курса группы 2125

Припадчев Артём

Проверил:

Зинчик А.А.

Санкт-Петербург

2014

**Задание:** требуется найти минимальный остов графа, то есть минимальное по весу поддерево графа G, содержащее все его вершины. Использовать алгоритмы Борувки и Прима.

Провести эксперимент на основе следующих данных: n=104+1, m=0,….,107 с шагом 105, q=1, r=106. Нарисовать полученные графики времени выполнения от количества ребер.

**Описание алгоритма**

 **Алгоритм Прима**

Построение начинается с дерева, включающего в себя одну (произвольную) вершину. В течение работы алгоритма дерево разрастается, пока не охватит все вершины исходного графа. На каждом шаге алгоритма к текущему дереву присоединяется самое лёгкое из рёбер, соединяющих вершину из построенного дерева и вершину не из дерева.

 Время работы 

 **Алгоритм Борувки**

 Работа алгоритма состоит из нескольких итераций, каждая из которых состоит в последовательном добавлении рёбер к остовному лесу графа, до тех пор, пока лес не превратится в дерево, то есть, лес, состоящий из одной компоненты связности.

В псевдокоде, алгоритм можно описать так:

1. Изначально, пусть  — пустое множество ребер (представляющее собой остовный лес, в который каждая вершина входит в качестве отдельного дерева).
2. Пока  не является деревом (что эквивалентно условию: пока число рёбер в  меньше, чем , где  — число вершин в графе):
	* Для каждой компоненты связности (то есть, дерева в остовном лесе) в подграфе с рёбрами , найдём самое дешёвое ребро, связывающее эту компоненту с некоторой *другой* компонентой связности. (Предполагается, что веса рёбер различны, или как-то дополнительно упорядочены так, чтобы всегда можно было найти единственное ребро с минимальным весом).
	* Добавим все найденные рёбра в множество .
3. Полученное множество рёбер  является минимальным остовным деревом входного графа.

Общее время работы алгоритмы составляет 

**Результат работы**



**Вывод:** экспериментальные данные тестирования алгоритмов Борувки и Прима подтвердили теоретические. Алгоритм Борувки действительно работает быстрее алгоритма Прима.