«Санкт-Петербургский государственный университет»

Факультет «Прикладная математика-процессы управления»

Отчет

по дисциплине: **«Алгоритмы и анализ сложности»**

на тему: **Эмпирический анализ алгоритма**

**“Алгоритм Борувки нахождения минимального**

**остовного дерева в графе”**

Выполнил: Малиновский Илья Владимирович,

Студент группы 19.Б-13

Санкт-Петербург, 2021

**Содержание**

1. Краткое описание алгоритма
2. Математический анализ алгоритма
3. Описание характеристик входных данных
4. Описание способа генерации входных данных
5. Программа, реализующая алгоритм
6. Вычислительный эксперимент
7. Анализ полученных данных
8. Характеристики вычислительной среды
9. Список используемой литературы

**Краткое описание алгоритма**

Алгоритм Борувки – это жадный алгоритм, разработанный и опубликованный Отакаром Борувкой, чешским математиком, наиболее знаменитым своей теорией графов.

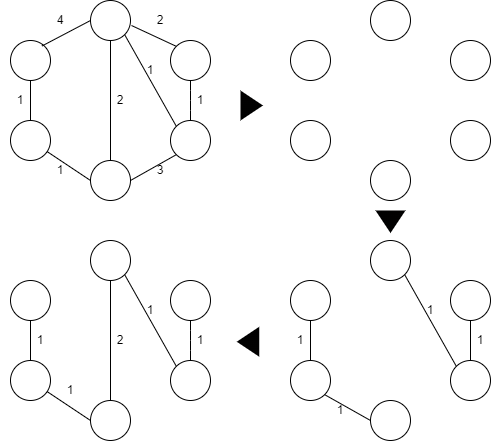
В этом алгоритме стоит отметить то, что это самый старый из известных алгоритмов для поиска минимального остовного дерева. Борувка придумал его в 1926 году, еще до того, как появились компьютеры в том виде, в каком мы их знаем сегодня. Он был опубликован как метод построения эффективной электросети.

Идея алгоритма довольно проста и интуитивно понятна. Разобьём его на несколько шагов:

1. Инициализируем каждую из вершин как отдельную компоненту.
2. Создадим минимальное остовное дерево **S** как пустое множество (в будущем содержащее решение).
3. Если компонент более одной

* Найдем для каждой минимальное по весу ребро, которое соединяет ее с другой компонентной.
* Если это ребро не в **S**, добавим его.

1. Если остался всего одна компонента – минимальное остовное дерево построено.



На рассмотренном выше примере, первым шагом граф разбился на пять компонент (по числу вершин). Далее, для каждой из компонент было найдено самое дешевое ребро, тем самым образовалось две компоненты. Последним шагом компоненты объединились в одну, добавлением ребра с весом 2 (как самым дешевым для обоих компонент).

**Математический анализ алгоритма.**

Алгоритм при каждом шаге уменьшает количество компонент связности в два раза. Изначально количество компонент равно числу вершин, следовательно, алгоритм совершит шагов.

Поскольку на каждом шаге проверяется ребер, то внутренний цикл имеет асимптотику .

Окончательная же временная сложность составляет .

**Описание характеристик входных данных.**

На вход

**Источники**

1. [Алгоритм Борувки — Алговики (algowiki-project.org)](https://algowiki-project.org/ru/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%91%D0%BE%D1%80%D1%83%D0%B2%D0%BA%D0%B8)