Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Дискретная математика

Лабораторная работа № 2

Тема: «Нахождение минимального остовного дерева методами Прима и Краскала»

Выполнил: студент группы ИВТ-22-1б

Игошев Матвей Иванович

Проверил: Ст. Преподаватель кафедры ИТАС

Рустамханова Г.И.

г. Пермь – 2024

Оглавление

Цель работы 3

Задачи работы 4

Этапы выполнения 5

Заключение 6

Список используемой литературы 7

# Цель работы

Создать консольное приложение, которое будет находить остов минимального веса согласно методам Прима и Краскала.

# Задачи работы

1. Изучить теорию связанную с алгоритмами Прима и Краскала.
2. Реализовать вспомогательные функции.
3. Реализовать основные функии для получения остовов.
4. Протестировать приложение

# Этапы выполнения

**Теоретическая часть**

Для того, чтобы перейти к реализации консольного приложения нужно понять что такое остов. Остовным деревом (остовом) связного графа называется любой его остовный подграф, являющийся деревом.

В рамках алгоритмов Прима и Краскала находится минимального веса остов(минимальное остовное дерево). Алгоритм Прима предлагается находить минимальное остовное дерево с помощью способа ближайшего соседа. Первое — выбираем самое малое ребро, второе — добавляем минимальные соседние ребра к последнему добавленному, третье — останавливаемся, при включении всех вершин графа.

Алгоритм Краскала работает немного иначе. Составляется список ребер, позже сортируются по весу. Дальше необходимо добавлять ребра таким образом, чтобы не образовывались циклы. Останавливаемся, когда все вершины будут задействованы. Для правильной реализации алгоритма необходимо использовать систему непересекающихся множеств для проверки на образование циклов.

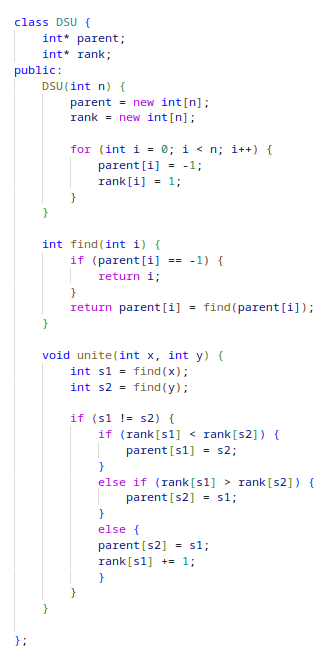
Система непересекающихся множеств является конечным множеством, которое разбито на непересекающиеся подмножетва (классы). Данная система поддеживает операции makeset, union, find.

**Объяснение работы кода программы**

Для реализации приложения был написан класс Graph:

Класс содержит ряд функций для функционирования приложения таких как: getVerts, setMatrix, printvec, printMtx, а также конструкторы класса и функции необходимые для нахождения остова методами Прима и Краскала. Рассмотрены в рамках отчета будут не все функции, но только те, что позволяют выполнить алгоритмы Прима и Краскала.

Как было упомянуто ранее алгоритм Краскала был реализован, при использовании системы непересекающихся множеств. Реализована она была следующим образом:

****

Реализована была система в виде класса с двумя методами unite и find. Реализация была выполнена на основе дерева.

Алгоритмы были реализованы в виде методов класса Graph. Алгоритм Прима выглядит следующим образом:

Создается вектор для хранения обработанных вершин, после выполняется цикл. В рамках цикла выполняется нахождение очередного минимального ребра, оно добавляется в возвращаемое дерево и также вершина ребра добавляется в список использованных. Цикл повторяется до тех пор пока не будет использована каждая вершина.

Алгоритм Краскала реализован следующим образом:

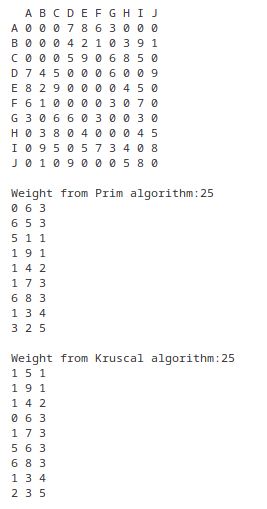


В рамках алгоритма создаются два вектора edges\_values, edges\_indices. Такое разделение необходимо для сортировки. Для сортировки используется сортировка подсчетом. После создается объект системы непересекающихся множеств. Далее выполняется цикл, в рамках которого выполняется идет добавление ребер, если при их добавлении не образуется цикл. После новое дерево возвращается.

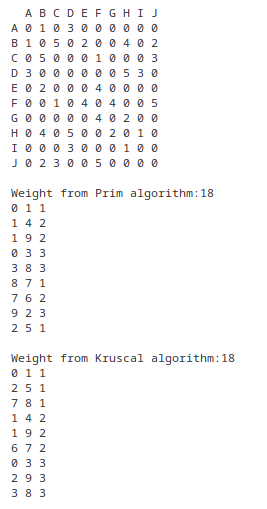
**Тестирование программы**

Для тестирования программы было подобрано несколько подобрано несколько графов.

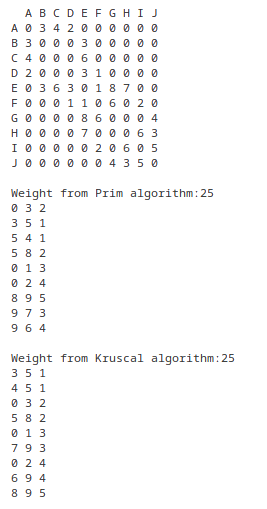
Результат тестирования первого графа:



Результат тестирования второго графа:



Результат тестирования третьего графа:



# Заключение

В ходе работы были закреплены знания об остовах, минимальных остовных деревьях и о методах их нахождения. Были реализованы алгоритмы нахождения остовов.

# Список используемой литературы

1. [https://habr.com/ru/articles/569444/](https://ru.wikipedia.org/wiki/Метод_Куайна)
2. https://cp-algorithms.com/data\_structures/disjoint\_set\_union.html