Московский авиационный институт

(Национальный исследовательский университет)

институт “Информационные технологии и прикладная математика”

кафедра “Математическая кибернетика”

КУРСОВАЯ РАБОТА

*по курсу* «Дискретная математика»

*2-й семестр*

|  |  |
| --- | --- |
| Тема: | Теория графов, алгебраические структуры, теория алгоритмов.  Построение минимального остовного дерева методом Прима. |

|  |  |
| --- | --- |
| *Студент*: | Данил Сергеевич  Бонокин |
| *Группа*: | М8О-110Б |
| *Руководитель*: | Н.С. Алексеев |
| *Оценка*: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| *Дата*: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

### 

### Введение

В настоящем отчете по курсовой работе приведены результаты, полученные в рамках изучения курса “Дискретная математика” во втором учебном семестре.

В части I приведены решения типовых задач по теории графов, а также представлена реализация алгоритма построение минимального остовного дерева методом Прима. Текст программы и тестовые примеры вынесены в Приложение.

В части II приведены решения типовых задач по темам “Алгебраические структуры” и “Теория алгоритмов”.

### Часть I. Теория графов

*Здесь должны содержаться постановки и решения задач А, Б, 1-7.*

### Часть II. Теория алгоритмов и алгебраические структуры

*Здесь должны содержаться постановки и решения задач 7-10.*

### 

### Часть III. Программная реализация алгоритма построения минимального остовного дерева методом Прима

#### Описание алгоритма.

В алгоритме Прима на каждом шаге рассматривается частичное решение задачи, представляющее собой дерево. Вначале это дерево состоит из единственной вершины, в качестве которой может быть выбрана любая вершина графа. Затем к дереву последовательно добавляются ребра и вершины, пока не получится остовное дерево. Для того чтобы из текущего дерева при добавлении нового ребра опять получилось дерево, новое ребро должно соединять вершину дерева с вершиной, еще не принадлежащей дереву. Такие ребра называются подходящими. В алгоритме Прима применяется следующее правило выбора: на каждом шаге из всех подходящих ребер выбирается ребро наименьшего веса. Это ребро вместе с одной новой вершиной добавляется к дереву.

#### Описание реализации алгоритма

Заведем переменную no\_edge = 1, в которой будем хранить количество вершин, которые мы посетили, а также массив Visit с длинной равной количеству вершин, который будет отображать вершины, которые мы посетили. К каждому значению присвоим False, а к первой вершине присвоим True. Потом на каждом шагу будем проходить по каждому ребру, у которого одна из вершин уже посещена, а вторая нет, и если это ребро имеет минимальный вес из всех таких вершин, то в массиве Visit помечаем вершины этого ребра True. На каждом шаге прибавляем к no\_edge 1. Цикл работает до тех пор пока no\_edge не равен количеству вершин в изначальном графе.

#### Описание программы и инструкции по работе с ней

В файле main.py происходит работа с файлом.

Установить программу можно с моего GitHub репозитория при помощи команды:

* *git clone https://github.com/Bonik0/kp-Discra*

Далее при помощи приложения ГРАФОИД (инструкция к нему также лежит внутри репозитория) необходимо построить граф. В качестве исполняемого файла необходимо выбрать:

* *~\ kp-Discra\ main.exe*

#### Оценка вычислительной сложности алгоритма

Так как мы на каждом шагу ищем ребро с наименьшим весом и повторяем этот процесс, пока не посетим все вершины, то сложность алгоритма равна где m – количество вершин, а n – количество ребер. А так как мы работаем с матрицей смежности, то итоговая сложность равна .

#### Результаты тестирования программы

#### Результаты тестирования программы вынесены в Приложение 2. Алгоритм работает корректно для любых графов.

#### Пример прикладной задачи

Алгоритм Прима имеет достаточно большую область применения, например у нас есть несколько городов, между которых необходимо построить дороги, так чтобы из любого города в любой другой, можно было попасть за минимальное время.

### Заключение

В ходе выполнения курсовой работы были решены 12 задач по различным разделам курса “Дискретная математика”.

Кроме того был изучен вопрос о различных методах поиска путей и маршрутов в графах. Была написана и отлажена программа, реализующая построение минимального остовного дерева методом Прима. Программа написана на языке программирования Python. Программа обеспечивает связь по установленному формату с системой ГРАФОИД, разработанной на кафедре 805, что дает возможность обеспечить графический интерфейс при ее использовании. Эта программа является основным результатом курсового проектирования.

## 

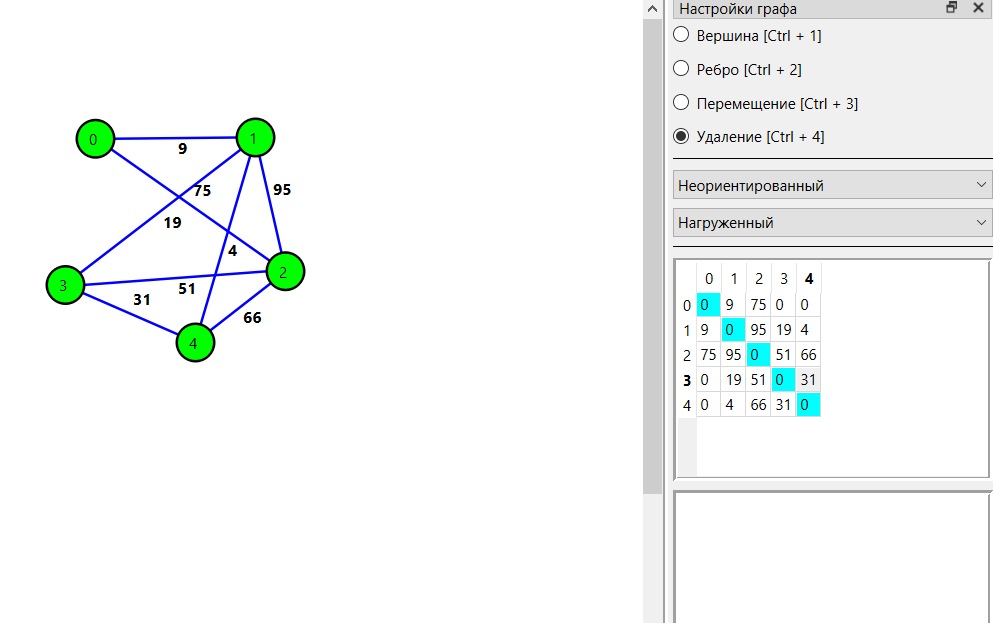
## **Список использованных источников**

* 1. *В.А. Таланов, В.Е. Алексеев.* Графы и алгоритмы. <https://www.intuit.ru> .
* 2. *Нефедов В.Н., Осипова В.А.* Курс дискретной математики. МАИ, 1992
* 3. *Нефедов В.Н.* Дискретные задачи оптимизации. <https://goo.gl/faUEEU>

### Приложение 1. Текст программы

### 

### Приложение 2. Результаты тестирования программы

* ****

