



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ Информатика и системы управления (ИУ)

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7 **«ОБРАБОТКА ГРАФОВ»**

Студент, группа

Артемьев И.О, ИУ7-33Б

2020 г.

Описание условия задачи

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

Техническое задание

Задана система двусторонних дорог, где для любой пары городов есть соединяющий их путь. Найти город с минимальной суммой расстояний до остальных городов.

Входные данные:

1. Файл с данными

Выходные данные:

1. Графическая визуализация всех путей из городов.
2. Графическая визуализация путей из города с минимальной суммой расстояний до остальных городов.

Функции программы:

1. Заполнить данные из файла
2. Найти город с минимальной суммой расстояний до остальных городов
3. Вывести время и объем памяти, затраченные на поиск города
1. Выйти

Обращение к программе: запускается из терминала командой:
./app.exe.

Аварийные ситуации:

1. Некорректный ввод пункта меню.
На входе: число не из меню или буква.
На выходе: сообщение «Ошибка: неверный пункт меню»
2. Некорректное названия файла, содержащего данные.
На входе: неправильное название файла.
На выходе: сообщение «Ошибка: не удалось открыть файл»

Структуры данных

Структура графа

```
typedef struct graph
{
    int size;
    int **matrix;
    int **paths;
} graph_t;
```

Поля структуры:

size — кол-во вершин в графе.

matrix — матрица смежности.

paths — матрица, содержащая кратчайшие пути из разных вершин графа.

Алгоритм

Для поиска кратчайшего пути используется алгоритм Дейкстры. Он подходит под условие задачи, так как в графе не может быть отрицательных весов. Алгоритм Дейкстры не подходит для несвязного графа, и так же не подходит для графа, где есть отрицательные веса у дуг. Но, так как в моей задаче таких случаев быть не может, то данный алгоритм идеально подходит для решения моей задачи.

Тесты

	Тест	Пользовательский ввод	Результат
1	Некорректный ввод пункта из меню	-1	Ошибка: неверный пункт меню
2	Некорректный ввод пункта из меню	A	Ошибка: неверный пункт меню
3	Некорректный ввод названия файла	IU7	Ошибка: неверный пункт меню
4	Некорректный ввод названия файла	123	Ошибка: неверный пункт меню
5	Корректный ввод названия файла	Graph.txt	Графическая визуализация графов

Оценка эффективности

Время выполнения (в тактах процессора):

Количество вершин	Время выполнения
5	3265
10	15655
15	73356

Занимаемая память (в байтах):

Количество вершин	Занимаемая память
5	224
10	824
15	1824

***тип хранения – матрица смежности**

Контрольные вопросы

1. Что такое граф?

Граф – конечное множество вершин и соединяющих их ребер; $G = \langle V, E \rangle$. Если пары E (ребра) имеют направление, то граф называется ориентированным; если ребро имеет вес, то граф называется взвешенным.

2. Как представляются графы в памяти?

С помощью матрицы смежности или списков смежности.

3. Какие операции возможны над графами?

Обход вершин, поиск различных путей, исключение и включение вершин.

4. Какие способы обхода графов существуют?

Обход в ширину (**BFS – Breadth First Search**), обход в глубину (**DFS – Depth First Search**).

5. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи, необязательно иерархические.

6. Какие пути в графе Вы знаете?

Эйлеров путь, простой путь, сложный путь, гамильтонов путь.

7. Что такое каркасы графа?

Каркас графа – дерево, в которое входят все вершины графа, и некоторые (необязательно все) его рёбра.

Вывод

Была реализована программа обработки графа и нахождения минимальной суммы путей от одной вершины графа к другим.

Для реализации программы был использован алгоритм Дейкстры (при 15 вершинах 73356 тактов), так как именно он находит кратчайшие пути от одной из вершин графа до всех остальных. Сам граф хранится, как матрица смежности, потому что так более эффективно по памяти (при 15 вершинах всего 1824 байта). Эта реализация была выбрана для удобства реализации алгоритма Дейкстры, а также для удобства получения матрицы кратчайших путей, которая по размерности совпадает с матрицей смежности.