



**Министерство науки и высшего образования 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

КАФЕДРА

Программное обеспечение ЭВМ и информационные технологии

**Отчет по лабораторной работе №8
«ГРАФЫ»**

Студент

Равашдех Фадей Хешамович

Группа

ИУ7 – 35Б

Преподаватель

Никульшина Т.А.

Оглавление

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	3
<u>ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....</u>	3
<u>ПРИМЕР РАБОТЫ ПРОГРАММЫ.....</u>	3
<u>ОПИСАНИЕ СТРУКТУР ДАННЫХ.....</u>	5
<u>ОПИСАНИЕ ФУНКЦИЙ ПРОГРАММЫ.....</u>	6
<u>ОПИСАНИЕ АЛГОРИТМА.....</u>	7
<u>НАБОР ТЕСТОВ.....</u>	7
<u>ОЦЕНКА ЭФФЕКТИВНОСТИ (СЕКУНД).....</u>	8
<u>ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	8

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Предложить вариант реальной задачи, для решения которой можно использовать разработанную программу. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

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

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Входные данные: Номер пункта меню, строка пути к файлу с данными.

Выходные данные: Графическая визуализация графа, время работы алгоритма, суммы расстояний от каждого города до остальных, наименьшая сумма расстояний и расстояния, из которых получается минимальная сумма.

Обращение к программе: Запускается через терминал командой: ./app.exe.

Функции меню программы:

0. Информация о программе
1. Считать данные из файла
2. Найти кратчайшие пути алгоритмом Дейкстры
3. Найти кратчайшие пути алгоритмом Флойда-Уоршелла
4. Запись в файл
5. Выйти

Аварийные ситуации:

1. Ошибка выделения памяти;
2. Прекращение ввода (EOF);
3. Некорректный ввод;
4. Некорректный выбор действия меню программы;
5. Пустая строка;
6. Некорректный файл.

ПРИМЕР РАБОТЫ ПРОГРАММЫ

Есть файл ./data/test, в котором хранится матрица смежности графа в координатном виде:

6 6 9

0 1 7

0 2 9
0 5 14
1 2 10
1 3 15
2 3 11
2 5 2
3 4 6
4 5 9

Запуск программы, чтение файла и нахождение кратчайших путей:

\$./app.exe

```
+-----+  
|               МЕНЮ  
+-----+  
| 0 | Информация о программе  
| 1 | Считать данные из файла  
| 2 | Найти кратчайшие пути алгоритмом Дейкстры  
| 3 | Найти кратчайшие пути алгоритмом Флойда-Уоршелла  
| 4 | Запись в файл  
| 5 | Выйти  
+-----+
```

Введите пункт меню: 1

Введите путь к файлу с данными (например: 'data/10'):

data/test

Файл успешно прочитан.

```
+-----+  
|               МЕНЮ  
+-----+  
| 0 | Информация о программе  
| 1 | Считать данные из файла  
| 2 | Найти кратчайшие пути алгоритмом Дейкстры  
| 3 | Найти кратчайшие пути алгоритмом Флойда-Уоршелла  
| 4 | Запись в файл  
| 5 | Выйти  
+-----+
```

Введите пункт меню: 2

Сумма от 0 вершины = 67

Сумма от 1 вершины = 65

Сумма (наименьшая) от 2 вершины = 43: 9 10 0 11 11 2

Сумма от 3 вершины = 65

Сумма от 4 вершины = 67

Сумма от 5 вершины = 47

Алгоритм Дейкстры работает за время: 0.000029 секунд

Запись матрицы в файл:

```
+-----+  
|               МЕНЮ  
+-----+  
| 0 | Информация о программе  
| 1 | Считать данные из файла
```

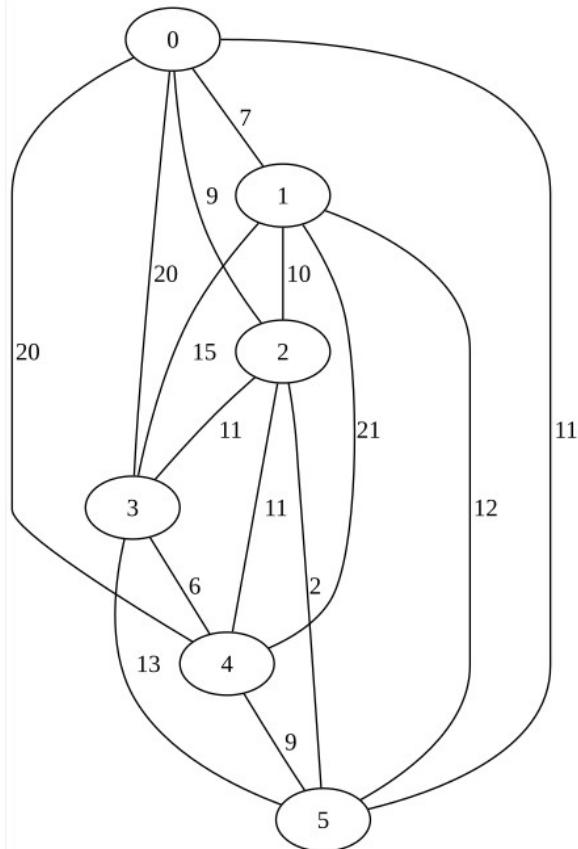
```
| 2 | Найти кратчайшие пути алгоритмом Дейкстры
| 3 | Найти кратчайшие пути алгоритмом Флойда-Уоршелла
| 4 | Запись в файл
| 5 | Выйти
```

```
+-----+
Введите пункт меню: 4
```

Перевод из формата .dot в формат .svg:

```
$ dot -Tsvg res.dot > res.svg
```

Получившиеся изображение res.svg:



ОПИСАНИЕ СТРУКТУР ДАННЫХ

Описание структуры графа (матрица смежности):

```
struct graph
{
    matrix_t matrix;
};
```

Описание структуры матрицы:

```
typedef struct
{
    size_t **matrix;      // Массив указателей на строки матрицы
    size_t n, m;          // Количество строк и столбцов соответственно
} matrix_t;
```

ОПИСАНИЕ ФУНКЦИЙ ПРОГРАММЫ

```
// Функция-контроллер
int controller(int option, graph_t *graph, char **filename);
// Основная функция
int main(void);
```

Функции графа

```
// Чтение из файла
int graph_read(char **filename, graph_t *graph);
// Запись в файл
int graph_write(graph_t graph);
// Обертка над алгоритмом
void wrapper(graph_t graph);
// Деструктор
void graph_del(graph_t *graph);
```

Функции матрицы

```
// Чтение матрицы из файла
int read_matrix(char *filename, matrix_t *matrix);
// Освобождение матрицы
void free_matrix(matrix_t *matrix);
```

Функции ввода/вывода

```
// Вывод меню программы
void print_menu(void);
// Вывод информации о программе
void print_info(void);
// Чтение строки неопределен. длины из стандартного потока ввода
int get_input(char **str, FILE *f);
// Чтение строки определенной длины из стандартного потока ввода
int get_n_input(char pch[], size_t n);
// Получение выбранного пользователем пункта
int get_option(size_t *option);
// Вывод сообщения, поясняющего ошибку
void print_error_msg(int rc);
// Вывод сообщения, поясняющего ошибку, и возврат того же кода
// возврата
int print_error(int rc);
```

ОПИСАНИЕ АЛГОРИТМА

1. Алгоритм подсчета минимальных расстояний между городами: алгоритм Флойда-Уоршелла или для каждого города запускается алгоритм Дейкстры.
2. Алгоритм Дейкстры: инициализируется два массива: массив минимальных расстояний от начальной вершины до остальных вершин (заполнен максимальным значением типа данных) и массив посещения вершин (заполнен нулями). Для каждой вершины: Выбирается ближайшая непосещенная вершина, отмечается в массиве посещений. Затем для каждой вершины смежной выбранной, проверяется меньше ли сумма расстояний от начальной до выбранной и от этой до выбранной, чем расстояние до этой вершины от начальной, если да записывается новое расстояние в массив расстояний.
3. Алгоритм Флойда-Уоршелла: если N — число вершин графа, то N раз проходится по верхней половине (т. к. граф неориентированный) матрицы смежности и для каждого ребра при наличии альтернативного двухреберного пути, его длина сравнивается с длиной этого ребра и, если он короче, то длина этого (и симметричного ему в нижней половине матрицы смежности) ребра приравнивается длине нового, более короткого.

НАБОР ТЕСТОВ

№	Название теста	Пользовательский ввод	Вывод
1	Некорректное действие	10	Некорректный номер пункта меню. Действия не существует. Ошибка: Действие с введенным пунктом отсутствует.
2	Некорректный ввод файла	1 \n	Ошибка: Некорректный ввод.
3	Работа с некорректным файлом	1 not_existing_file.txt 4	Ошибка: некорректный файл.
4	Некорректный ввод	1 data/10 \n	Ошибка: Некорректный ввод.
5	Корректный ввод	1 data/10	Файл успешно прочитан.
6	Корректное использование	1 data/10 2	Файл успешно прочитан. Алгоритм работает за время T . *суммы расстояний*

ОЦЕНКА ЭФФЕКТИВНОСТИ (СЕКУНД)

Кол-во элементов	Алгоритм Дейкстры	Алгоритм Флойда-Уоршелла
1	0 . 000004	0 . 000001
10	0 . 000090	0 . 000027
100	0 . 010506	0 . 006273
1000	2 . 636577	1 . 546023

Асимптотика у обоих алгоритмов равно $O(N^3)$, но алгоритм Флойда-Уоршелла быстрее алгоритма Дейкстры. Пусть s – размер типа (`size_t`) равен 8, N – количество элементов. Тогда дополнительная память алгоритма Дейкстры (2 массива): $2 * s * N$. Алгоритм Флойда-Уоршелла не требует дополнительной памяти, т.к. работает на исходной матрице смежности.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

- Что такое граф?** Граф – упорядоченная пара (V , E) из непустого множества V вершин и множества E рёбер. Если ребра имеют направление, то граф ориентированный, если имеют вес, то граф взвешенный.
- Как представляются графы в памяти?** С помощью матрицы смежности или массива списков смежности.
- Какие операции возможны над графиками?** Поиск кратчайшего пути от одной вершины к другой (если есть), поиск кратчайшего пути от одной вершины ко всем другим, поиск кратчайших путей между всеми вершинами, поиск эйлерова пути (если есть), поиск гамильтонова пути.
- Какие способы обхода графов существуют?** Обход в ширину — просмотр всех соседей, обход в глубину — проходит по кратчайшим ребрам.
- Где используются графовые структуры?** Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи. Интернет-сети, сети в электротехнике, энергетике, социальные графы и социограммы, навигационные карты.
- Какие пути в графике Вы знаете?** Эйлеров путь, гамильтонов путь, кратчайший путь, простой путь, сложный путь.
- Что такое каркасы графа?** Каркас графа – это подграф графа, состоящий из одного оственного дерева для каждой компоненты связности графа.

ВЫВОД

Алгоритм Флойда-Уоршелла работает быстрее, чем алгоритм Дейкстры для всех вершин, т. к. в случае неориентированного графа, нет смысла проходить через всю матрицу, а можно ограничиться только верхней половиной, т. к. значения нижней половины дублируют значения верхней.