



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

КАФЕДРА \_\_\_\_\_ Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**  
**“ТИПЫ И СТРУКТУРЫ ДАННЫХ”**  
**ЛАБОРАТОРНАЯ РАБОТА №7**

Студент \_\_\_\_\_ Ляпина Наталья Викторовна  
*фамилия, имя, отчество*

Группа \_\_\_\_\_ ИУ7-32Б

Вариант \_\_\_\_\_ 9

Студент \_\_\_\_\_ Ляпина Н.В.  
*подпись, дата* *фамилия, и.о.*

Преподаватель \_\_\_\_\_ Силантьева А.В.  
*подпись, дата* *фамилия, и.о.*

2021 г.

## **Оглавление**

<b>1)</b>	Условие задачи.....	<b>3</b>
<b>2)</b>	Схема программы.....	<b>4</b>
<b>3)</b>	Описание программы.....	<b>11</b>
<b>4)</b>	Текст программы.....	<b>15</b>
<b>5)</b>	Заключение.....	<b>18</b>
<b>6)</b>	Список литературы.....	<b>18</b>

# Задание

## 1. Общее задание

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

## 2. Задание по варианту

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

## 3. Входные данные

- Команда для выбора действия (от 0 до 5)
  1. Ввести граф вручную
  2. Ввести граф из файла
  3. Вывести граф
  4. Найти вершину с наименьшей суммой тупей до всех остальных вершин
  0. Выход из программы
- Количество вершин
- Вес ребер
- Имя файла

## ● Требования к входным данным

- 1) Валидная команда
- 2) Количество вершин натуральное число
- 3) Вес ребер не отрицательное число

## 4. Выходные данные

- Файл с графом dot
- Вершина с наименьшей суммой путей

## 5. Действие программы

Программа выполняет обработку графа

## 5. Обращение к программе

Программа может быть вызвана через консоль компилятора с помощью команды  
“./app.exe”

## 6. Аварийные ситуации

В случае аварийной ситуации выводится сообщение об определенной ошибке.

### *Неверный ввод:*

- Ошибка при вводе команды
- Ошибка при вводе количества вершин
- Ошибка при вводе веса ребер

## Структура данных

### Граф

```
typedef struct
```

```
{
```

```
    int matrix[MAX_SIZE][MAX_SIZE];
```

```
    int size;
```

```
} graph_t;
```

```
MAX_SIZE = 500
```

size - количество вершин графа

### Структура для алгоритма Дейкстры

```
typedef struct
```

```
{
```

```
    int min_len[MAX_SIZE];
```

```
    int ver [MAX_SIZE];
```

```
} dextra_t;
```

min\_len - минимальное расстояние до каждой вершины

ver - посещенные вершины

## Интерфейс модулей

*Для обработки типа `graph_t` используются функции:*

```
int read_graph(graph_t *table)
```

// Функция чтения графа с клавиатуры

// table - граф

```
int read_graph_by_file(graph_t *table, char *file_name)
```

// Функция чтения графа из файла

// table - граф, file\_name - имя файла

```
int print_graph(graph_t table, char *file_name, int min_versh)
```

// Функция печати графа

// table - граф, file\_name - имя файла, min\_versh - вершина с мин. суммой

```
int dextra_alg(graph_t *table, dextra_t *argv, int versh)
```

// Алгоритм Дейкстры

// table - граф, argv - аргументы, versh - вершина относительно которой происходят вычисления

## Описание алгоритма

1. Вводится команда для выполнения определенной функции программы.
2. Выполняется введенная функция
  - Ввод графа с клавиатуры
  - Ввод графа из файла
  - Печать графа
  - Поиск вершины с минимальной суммой путей
    - Для поиска вершины с минимальной суммой путей используется алгоритм Дейкстры для каждой вершины графа

- Результат работы алгоритма сравнивается с минимумом и находится новый минимум.
3. При ошибке выполнения функции выводится сообщение об ошибке и программа завершается с ненулевым кодом возврата.

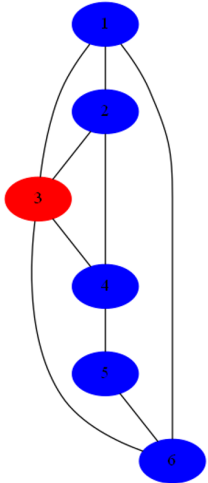
## Тесты

### *Негативные тесты*

№	Описание	Вводимая структура	Результат
1	Недопустимый символ команде меню	a2	Ошибка
2	Несуществующая команда меню	10	Ошибка
3	Недопустимое количество вершин	-2	Ошибка
4	Несуществующий файл	123.ttt	Ошибка
5	Недопустимое значение веса ребра	1.3	Ошибка

### *Позитивные тесты*

№	Описание	Входные данные	Результат
1	Правильный пункт меню	comand = 1	Выбор действия
2	Правильный файл	test.txt	Успешное чтение файла
3	Правильное количество вершин	2	Успешное чтение
4	Правильный вес ребра	3	Успешное чтение

5	Ввод графа из файла	6 $1 - 2 = 7$ $1 - 3 = 9$ $1 - 6 = 14$ $2 - 3 = 10$ $2 - 4 = 15$ $3 - 4 = 11$ $3 - 6 = 2$ $4 - 5 = 6$ $5 - 6 = 9$	
---	---------------------	--	---

## ***Вывод***

Для хранения данных графа была выбрана матрица смежностей. Так как в данном графе система двухсторонних дорог. Следовательно, наиболее удобная структура данных - матрица.

Для поиска вершины с наименьшей суммой путей был выбран алгоритм Дейкстры, поскольку выбранная структура данных подходит для этого алгоритма, а также он наиболее простой и понятный. Также алгоритм Дейкстры удобен тем, что решает задачу полностью, без использования других алгоритмов.

## ***Ответы на контрольные вопросы***

### **1. 1. Что такое граф?**

Граф – конечное множество вершин и ребер, соединяющих их:  $G = \langle V, E \rangle$ , где  $V$  – конечное непустое множество вершин;  $E$  – множество рёбер (пар вершин).

### **2. Как представляются графы в памяти?**

Графы в памяти могут представляться различным способом. Один из видов представления графов – это матрица смежности  $B (n \times n)$ ; В этой матрице элемент  $b[i, j] = 1$ , если ребро, связывающее вершины  $V_i$  и  $V_j$  существует и  $b[i, j] = 0$ , если ребра нет. У неориентированных графов матрица смежности всегда симметрична. Во многих случаях удобнее представлять граф в виде так называемого списка смежностей. Список смежностей содержит для каждой вершины из множества вершин  $V$  список тех вершин, которые непосредственно связаны с этой вершиной. Каждый элемент ( $ZAP[u]$ ) списка смежностей является записью, содержащей данную вершину и указатель на следующую запись в списке (для последней записи в списке этот

указатель – пустой). Входы в списки смежностей для каждой вершины графа хранятся в таблице (массиве) (BEG [u])

### **3. Какие операции возможны над графами?**

- поиск кратчайшего пути от одной вершины к другой (если он есть);
- поиск кратчайшего пути от одной вершины ко всем другим;
- поиск кратчайших путей между всеми вершинами;
- поиск эйлера пути (если он есть);
- поиск гамильтонова пути (если он есть);
- исключение/включение вершины/ребра

### **4. Какие способы обхода графов существуют?**

Обходы в глубину, в ширину.

### **5. Где используются графовые структуры?**

Графы нашли применение практически во всех отраслях научных знаний: физике, биологии, химии, математике, истории, лингвистике, социальных науках, технике и т.п. Наибольшей популярностью теоретико-графовые модели используются при исследовании коммуникационных сетей, систем информатики, химических и генетических структур, электрических цепей и других систем сетевой структуры.

### **6. Какие пути в графе Вы знаете?**

- Эйлеров путь (произвольный путь в графе, проходящий через каждое ребро графа точно один раз)
- Простой путь (путь в графе, проходящий через каждую вершину однократно)
- Гамильтонов путь (путь в графе, проходящий в точности один раз через каждую вершину графа (а не каждое ребро))

### **7. Что такое каркасы графа?**

Остовое дерево или каркас графа - это подграф графа, который содержит все вершины графа и является деревом