



**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 №7** **«ОБРАБОТКА ГРАФОВ»**

Студент

Цветков Иван Алексеевич

Группа

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.

## **ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ**

Задана система двусторонних дорог. Определить, можно ли, построив еще три новые дороги, из заданного города добраться до каждого из остальных городов, проезжая расстояние не более  $T$  единиц.

### **Входные данные:**

**1. Целое число, представляющее собой количество вершин графа (городов):**

целое число в диапазоне от 2 до 10

**2. Дополнительный ввод:** поле типа `int` отвечающие за максимальный путь, город, из которого производится поиск и длины дорог между городами

### **Выходные данные:**

1. Результат выполнения команды
2. Сообщение об ошибке (при ее возникновении)

### **Функции программы:**

1. Ввести данные
2. Решение поставленной задачи
3. Вывод результата в консоль и в качестве графов с использованием GraphViz

(все выполняется последовательно при запуске программы)

### **Обращение к программе:**

Запускается через терминал командой ./app.exe

### **Аварийные ситуации:**

1. Неверно введено количество городов (вершин графа)  
(не число или число меньше 2 или больше 10)
2. Неверно введена длина дороги между городами  
(не число или число меньше 0)
3. Неверно введен начальный город  
(не число или число меньше 1 или больше количества городов)
4. Неверно введена максимальная длина пути в другой город  
(не число или число меньше 0)
5. Неверно задан граф — при добавлении путей он не связан

## **Описание структуры данных**

*Структура для хранения матрицы стоимостей для графа*

```
typedef struct mat_r
{
    int **data;

    int size;
} mat_t;
```

Поля структуры:

1. int \*\*data — матрица стоимостей введенного графа
2. int size — количество вершин графа

*Структура для хранения индексов матрицы стоимостей для отсутствующих дорог*

```
typedef struct no_roads_r
{
```

```
    int i;  
    int j;  
  
    int len;  
} no_roads_t;
```

Поля структуры:

1. int i — индекс строки матрицы стоимостей с пустой дорогой
2. int j — индекс столбца матрицы стоимостей с пустой дорогой
3. int len — количество пустых дорог в графе

## ОПИСАНИЕ АЛГОРИТМА

1. Происходит считывание количества городов (вершин графа), расстояний между городами, города, с которого идет поиск, а также максимальная длина пути из введенного города во все остальные
2. Затем идет решение самой задачи. Граф представляется в виде матрицы стоимостей (отсутствие пути обозначено 0). Затем находятся все отсутствующие пути в графе. После запускается полный перебор всех возможных троек новых дорог (минимальной длины - 1), одновременно с

этим для каждой комбинации путей выполняется алгоритм Дейкстры для нахождения кратчайших путей из заданной вершины во все остальные. Цикл продолжается до тех пор, пока не будет найдена такая тройка новых дорог, при которой можно будет добраться до каждого города из заданного по не большему расстоянию, чем заданное, или пока не будут перебраны все возможные комбинации

3. Затем на экран выводятся, если были найдены, пути, по которым содержится минимальное расстояние до каждого из городов, а также графы (входной и выходной) в виде рисунков с помощью GraphViz. Иначе выводится сообщение, что невозможно найти такие пути

## НАБОР ТЕСТОВ

	Название теста	Пользовательский ввод	Результат
1	Неверно введено количество городов (вершин графа)  (не число или	iu  или  11  или	Ошибка: неверно введено количество городов

	число меньше 2 или больше 10)	1	
2	Неверно введена длина дороги между городами  (не число или число меньше 0)	iu  или  -1	Ошибка: неверно введена метка пути. Повторите попытку
3	Неверно введен начальный город  (не число или число меньше 1 или больше количества городов)	iu  или  -1	Ошибка: неверно введен пункт отправления. Повторите
4	Неверно введена максимальная длина пути в другой город  (не число или число меньше 1)	iu  или  -1	Ошибка: неверно введена максимальная длина пути. Повторите
5	Неверно задан граф — при добавлении путей он не связан	Задано 5 городов без путей	Ошибка: граф не связан
6	Есть такая комбинация дорог, что удовлетворяет	Верные данные	Пути, по которым располагаются самые короткие по длине дороги до

	условию		города, а также картинки с исходным и выходным графами
7	Нет такой комбинации дорог, что удовлетворяет условию	Верные данные	Сообщение о невозможности найти такие пути при добавлении новых дорог, а также картинка с исходным графом

## Пример работы программы

Есть нужная комбинация:

Введите количество городов (вершин графа) (от 2 до 10): 5

Введите длину пути:

из 1 в 2: 6

из 1 в 3: 0

из 1 в 4: 6

из 1 в 5: 6

из 2 в 3: 4

из 2 в 4: 3

из 2 в 5: 0

из 3 в 4: 4

из 3 в 5: 6

из 4 в 5: 0

	( 1 )	( 2 )	( 3 )	( 4 )	( 5 )
( 1 )	-	6	0	6	6
( 2 )	6	-	4	3	0
( 3 )	0	4	-	4	6
( 4 )	6	3	4	-	0
( 5 )	6	0	6	0	-

Введите город, из которого будет производиться поиск: 1

Введите максимальную длину пути: 10

Кратчайшие пути в город 1 (не превышающие 10):

Из города 1 : 1

Из города 2 : 2 -> 3 -> 1

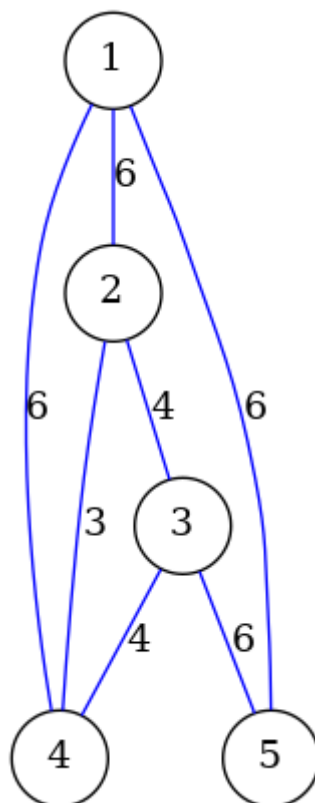
Из города 3 : 3 -> 1

Из города 4 : 4 -> 3 -> 1

Из города 5 : 5 -> 2 -> 3 -> 1

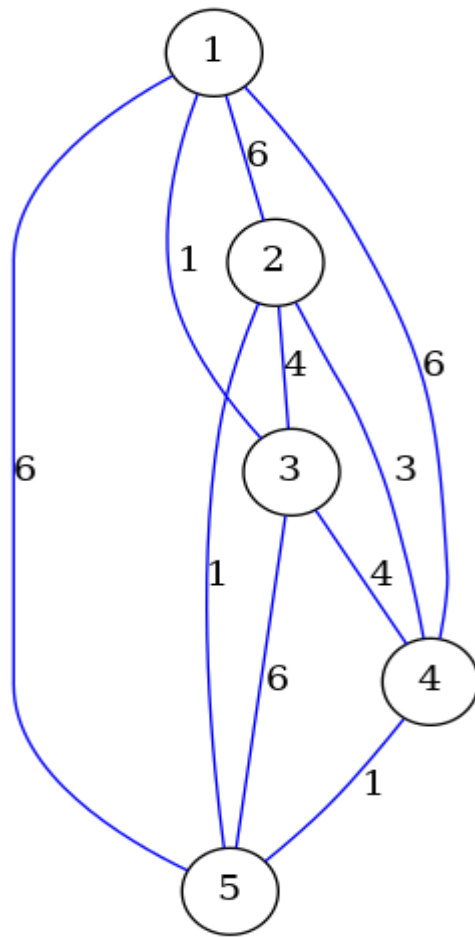
Найдено построение графа

Исходный граф:



Выходной граф:





Нет комбинации:

Введите количество городов (вершин графа) (от 2 до 10): 5

Введите длину пути:

из 1 в 2: 11

из 1 в 3: 0

из 1 в 4: 9

из 1 в 5: 0

из 2 в 3: 0

из 2 в 4: 12

из 2 в 5: 8

из 3 в 4: 9

из 3 в 5: 45

из 4 в 5: 9

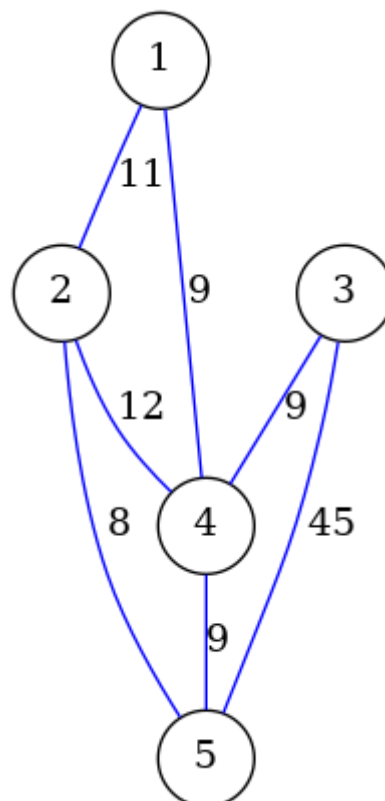
	( 1)	( 2)	( 3)	( 4)	( 5)
( 1)	-	11	0	9	0
( 2)	11	-	0	12	8
( 3)	0	0	-	9	45
( 4)	9	12	9	-	9
( 5)	0	8	45	9	-

Введите город, из которого будет производиться поиск: 1

Введите максимальную длину пути: 5

Нет возможности построить такой граф

Исходный граф:



# ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

## **1. Что такое граф?**

Граф –конечное множество вершин и соединяющих их ребер  
Обозначается  $G = \langle V, E \rangle$

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

С помощью матрицы смежности (стоимостей) или списка смежности

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

Обход вершин, их исключение и включение, поиск различных путей

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

Обход в ширину и обход в глубину

## **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, непростой путь, гамильтонов путь

## **7. Что такое каркасы графа?**

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

## **ВЫВОД**

При решении задачи граф представляется в виде матрицы стоимостей (отсутствие пути обозначено 0).

Матрица стоимостей выбирается, поскольку мы имеем длины дорог, которые мы и представляем в таком виде для дальнейшего использования в алгоритме Дейкстры. Сам алгоритм Дейкстры выбирается для поиска кратчайших расстояний из заданной вершины во все остальные, который подходит для задачи, поскольку расстояния между городами не могут быть отрицательными.

В начале находятся все отсутствующие пути в графе. После запускается полный перебор всех возможных троек новых дорог (минимальной длины - 1), одновременно с этим для каждой комбинации путей выполняется алгоритм Дейкстры. Цикл продолжается до тех пор, пока не будет найдена такая тройка новых дорог, при которой можно будет добраться до каждого города из заданного по не большему расстоянию, чем заданное, или пока не будут перебраны все возможные комбинации