|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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**

**«ГРАФЫ»**

Студент\_\_\_\_\_\_Чыонг Ван Хао\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа\_\_\_\_\_*ИУ7И-31Б\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Вариант 9

Принял : Силантьева А. В.

**Описание условия задачи**

**Техническое 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.

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

**Входные данные**

Текстовый файл — файл с матрицей смежности графа.

**Выходные данные**

1. Целое число — номер города — номер вершины графа, сумма расстояний от которой до всех других вершин минимальная .

2. Матрицы целых чисел — матрица смежности графа и матрица кратчайших расстояний.

3. Изображение графа в формате png — исходный граф и граф с кратчайшими расстояниями и выделенной вершиной

**Функция программы**

Программа принимает матрицу смежности, описывающую граф (систему двусторонних дорог), и находит вершину графа (город), сумма расстояний от которой до других вершин (городов) минимальна. Выводит матрицу смежности,матрицу кратчайших расстояний и визуализирует исходный граф и граф кратчайших путей с искомой вершиной.

**Обращение к программе**

Программа запускается из терминала в директории с проектом при помощи команды «./app.exe name.txt», где name.txt — имя файла с матрицей смежности графа.

**Возможные аварийные ситуации и ошибки пользователя**

1. Файл не существует.

На выходе сообщение: «ERROR READING FILE!.»

2. Неверно задано число городов.

Входные данные : не целое число или целое число, меньшее 3

На выходе сообщение: « The number of cities incorrectly!»

3. Ошибка выделения памяти.

На выходе сообщение: « ERROR ALLOCATE! »

4. Ввод отрицательного веса графа

Входные данные : целое отрицательное число.

На выходе сообщение: « ERROR INPUT! »

5. Обработка несвязного графа

На выходе сообщение: « The graph is not connected!»

**Внутренняя структура данных**

Для реализации очереди при помощи массива используется структура:

typedef struct

{

    int size;

    int \*\*matrix;

    int \*\*road;

} graph\_t;

Её поля:

int size — число вершин графа;  
int \*\*matrix — матрица смежности графа;  
int \*\*road— матрица кратчайших расстояний графа

**Функции программы**

int get\_graph(graph\_t \*graph, const char \*filename);

Описание: ввод графа из файла.

void init\_graph(graph\_t \*graph);

Описание: инициализация структуры, описывающей граф.

int graph\_create(graph\_t \*graph);

Описание: создание структуры, описывающей граф.

void free\_graph(graph\_t \*graph);

Описание: освобождение памяти из-под структуры, описывающей граф;

int Method\_Dijkstra(graph\_t \*graph, const int start);

Описание: алгоритм Дейкстры.

int get\_best\_way(graph\_t \*graph, int \*best\_index);

Описание: нахождение вершины с минимальной суммой расстояний до других городов.

void visual\_graph(int \*\*matrix, const int size, char \*name, char \*filename, int node);

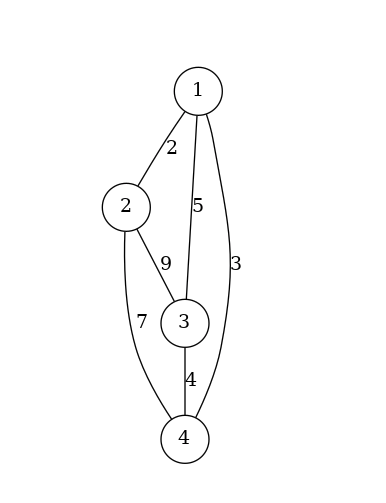
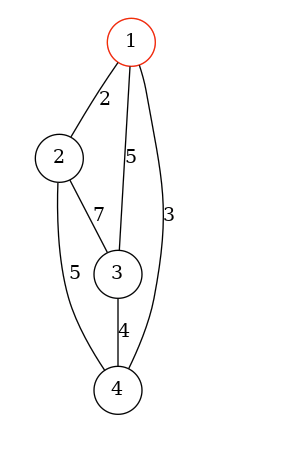
Описание: визуализация графа.

**Тесты**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Тест | Ввод | Вывод |
| **1** | Неверный ввод номера  веса ребра: не целое число | **z** | ERROR INPUT! |
| **2** | Неверный ввод числа городов: меньше 3 | **1** | The number of cities incorrectly! |
| **3** | Неверный ввод числа городов: не целое число | **c** | The number of cities incorrectly! |
| **4** | Ввод отрицательного веса графа | **-5** | ERROR INPUT! |
| **5** | обычный тест | 4  0 2 5 3  2 0 9 7  5 9 0 4  3 7 4 0 | Матрица смежности:  0 2 5 3  2 0 9 7  5 9 0 4  3 7 4 0  Матрица кратчайших расстояний:  0 2 5 3  2 0 7 5  5 7 0 4  3 5 4 0  Город с минимальной суммой расстояний до других городов: 1 |

**Примеры визуализации графов**

Begin Result

**Выводы**

В лабораторной работе граф представлен матрицей смежности, также граф  
можно представлять списком смежности. Для поиска кратчайших путей использован алгоритм Дейкстры. Выбор алгоритма обосновывается работой с графом только с положительным весами ребер и сложностью алгоритма O(n2) (например, у Флойда-Уоршелла О(n3)). Программу можно использовать для решение реальной задачи, например выбора населенного пункта для расположения центра скорой помощи, обслуживающей систему населенных пунктов.

**Контрольные вопросы**

Ответы на контрольные вопросы **1. Что такое граф?**Граф - это конечное множество вершин и ребер, соединяющих их, т. е.:  
G = < V,E >, где V – конечное непустое множество вершин; Е – множество ребер (пар вершин).  
**2. Как представляются графы в памяти?**Графы в памяти представляются при помощи матрицы смежности или при  
помощи списка смежности.  
**3. Какие операции возможны над графами?**Над графами возможны следующие операции: 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 эйлерова пути (если он есть), поиск гамильтонова пути (если он есть),  
оюход графа, исключение и включение вершин.  
**4. Какие способы обхода графов существуют?**Способы обхода графов: в глубину и в ширину.  
**5. Где используются графовые структуры?**Графовые структуры используются в задачах, где существуют произвольно  
связанные элементы. Например, в геоинформационных системах (ГИС).  
Существующие или вновь проектируемые дома, сооружения, кварталы и т. п.  
рассматриваются как вершины, а соединяющие их дороги, инженерные сети,  
линии электропередачи и т. п. — как рёбра. Применение различных  
вычислений, производимых на таком графе, позволяет, например, найти  
кратчайший объездной путь или ближайший продуктовый магазин,  
спланировать оптимальный маршрут.  
**6. Какие пути в графе Вы знаете?**Эйлеров путь - произвольный путь в графе, проходящий через каждое ребро  
графа точно один раз. При этом, если по некоторым вершинам путь проходит  
неоднократно, то он является непростым.  
Гамильтонов путь - путь в графе, проходящий в точности один раз через  
каждую вершину графа (а не каждое ребро).  
**7. Что такое каркасы графа?**Каркас графа - дерево, содержащие все вершины графа.