|  |  |
| --- | --- |
|  | **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)** |

Отчет по лабораторной работе № 7

«Графы»

Работу выполнил:

студент группы ИУ7-32Б

Лемешев А.П.  
Работу проверил:  
Никульшина Т.А.

Москва, 2021 г.

# Условие задачи

# Реализовать алгоритмы обработки графовых структур: поиск различных путей, проверку связности, построение остовых деревьев минимальной стоимости.

# Техническое задание

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 Т единиц.

**Входные данные**

- Граф:

- количество вершин.

- количество ребер.

- ребро в виде: *вершина1, вершина2, вес*.

- Номер вершины, из которой будет рассчитываться расстояние

- Максимальная длина пути

- Длина дорог, которые будут добавляться

**Выходные данные**

- Граф в файловом формате *graph.gv* для отображения в программе **graphviz.**

- Вывод: возможно или невозможно добраться из заданной вершины до всех остальных, не превышая заданного расстояния.

**Действие программы**

- Нахождение дорог, построив которые, можно будет из заданной вершины добираться до всех остальных, не превышая заданного расстояния.

**Обращение к программе**

Запуск программы производится через командную оболочку MSYS2 для Windows, терминал sh/bash для Linux. Запуск программы происходит с помощью команды:   
*./app.exe*.

**Возможные аварийные случаи**

1. Некорректный ввод графа
2. Некорректный ввод вершины
3. Некорректный ввод максимального пути
4. Некорректный ввод длину дорог, которые будут добавляться

# Структуры данных

int \*\*graph; *- граф, заданный матрицей смежности.*

int count\_node; *- количество вершин в графе.*

int count\_route; *- количество ребер в графе.*

int \*lens; *- массив пути*.

int \*f; *- массив посещённых вершин.*

**Описание алгоритма**

Граф считывается и строится в виде матрицы смежности. Выводится графическое изображение введенного графа. Далее проверяется, возможно ли добраться до всех вершин из заданной, не превышая заданное расстояние, не построив новых дорог. Расстояние из заданной вершины до всех остальных находится с помощью алгоритма **Дейкстры**. Если этого невозможно, то перебором всех возможных вариантов в граф сначала добавляется 1 ребро, затем 2, и потом 3. Если находится такой граф, что добраться до всех вершин становиться возможно, то программа выводит графическое изображение графа и выполнение программы прекращается.

*Существующие ребра не переписываются.*

# Вывод

Алгоритм **Дейкстры** был выбран из-за отсутствия отрицательных ребер в графе.

Преимущество представления графа как матрицу смежности над списком смежности:

- легкая для понимания структура.

- быстрая скорость обработки с большим количеством ребер.

- быстрая проверка существования ребра.

- в взвешенном графе список хранит два поля

Недостатки представления графа как матрицы смежности над списком смежности:

- занимает больше памяти, если граф малой связности или если граф

неориентированный, т.к. матрица получается симметричной.

# Результат работы программы

# 

# Ответы на контрольные вопросыы

**1. Что такое граф?**

Граф – конечное множество вершин и ребер, соединяющих их.

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

Матрица смежности, список смежностей.

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

Обход, поиск путей, поиск каркасов, поиск фундаментальных циклов, добавление и удаление вершин или ребер.

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

Обход в ширину, обход в глубину.

**5. Где используются грифовые структуры?**

Представление различный связанных данных, хранение и обработка картографической информации.

**6. Какие пути в графе Вы знаете?**

Простой путь – путь, в котором каждая из вершин графа встречается не более одного раза.

Контур – замкнутый путь в орграфе.

Эйлеров путь – проходит через каждое ребро ровно один раз.

Гамильтонов путь – проходит через каждую вершину ровно один раз.

**7. Что такое каркасы графа?**

Каркас графа – дерево, в которое входят все вершины графа, и некоторые ребра.