МГТУ им. Н. Э. Баумана

**Дисциплина типы и структуры данных**

**Лабораторный практикум №7**

**по теме: «Графы»**

**Вариант №3**

Работу выполнил:

студент группы ИУ7-35

Платонова Ольга

Работу проверил:

Москва, 2019 г.

***Цель работы***: реализовать алгоритмы обработки графовых структур:

* поиск различных путей
* проверку связности
* построение остовых деревьев минимальной стоимости.

***Условие задачи***

Найти самый длинный простой путь в графе. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных осуществить на усмотрение программиста. Результат выдать в графической форме.

***Техническое задание***

1. Организовать ввод графа по усмотрению программиста.

2. Представить граф любым способом, пригодным для обработки значение.

3. Найти максимально длинный простой путь в графе.

4. Вывести на экран граф.

*Входные данные:*

Название файла, содержащего описание графа.  
int scan\_from\_file(FILE \*filename, int \*\*matr, int n);

*Выходные данные*

Графическое представление графа.

void print\_matr(int \*\*matr, int n);

void output\_graph(FILE \*out, int \*\*matr, int n, int \*path, int p\_l);

*Структура данных*

*1. Матрица смежности*

int \*\*matr = alloc\_rows(n);

Динамическая структура.

*2. Массив пути*

int \*path = calloc(n, sizeof(int));

Динамическая структура.

***Описание алгоритма***

1. Считать из файла входные данные.

2. Записать данные в массив, прдеставляющий собой матрицу смежности.

3. Найти макисмальный простой путь:

осуществить обход графа в ширину

записать длину каждого узла в массив пути

найти макисмальный элемент вектора (массива)

int empty(int \*\*matr, int n);  
int max\_path(int \*\*matr, int n, int \*path, int \*p\_l);  
int traverse(int \*\*matr, int n, int num, int \*Q, int \*q\_l, int \*M, int \*path, int \*p\_l);

4. Вывести граф на экран, используя приложение.

***Тесты***

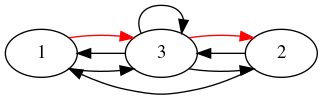
1. Пустой граф без ребер.

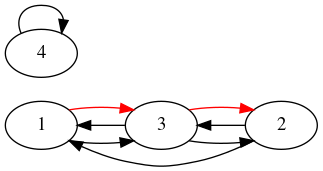
Graph cannot be built.

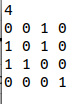
2. Смежный граф с петлями.

3. Граф, состоящий из одного узла.

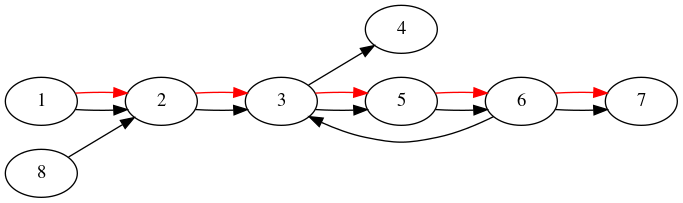
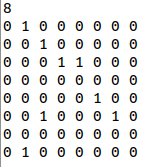
4.



5.



5.



***Вопросы***

*1. Что такое граф?*

Граф – это конечное множество вершин графа и набор ребер, то есть соединений между парами вершин.

*2. Как представляются графы в памяти?*

Графы в памяти могут представляться различным способом.

Один способ — матрица смежности B(n \* n). У неориентированных графов матрица смежности всегда симметрична.

Другой способ — список смежностей.

*3. Какие операции возможны над графами?*

* обход графа
* добавление/удаление вершин
* 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)

*4. Какие способы обхода графов существуют?*

Обход в глубину, обход в ширину.

*5. Где используются графовые структуры?*

Графовые структуры широко распространены в базе данных; они предназначены для представления различных связанных данных; хранения и обработки картографической информации.

*6. Какие пути в графе Вы знаете?*

Простой путь: каждая из вершин графа встречается не более одного раза (может вообще не встречаться).

Контур: замкнутый путь в ориентированном графе.

Эйлеров путь: каждое ребро встречается ровно один раз.

Гамильтонов путь: каждая вершина встречается ровно один раз.

*7. Что такое каркасы графа?*

Каркас графа — дерево, или в общем случае лес, в котором содержатся все вершины графа и некоторые ребра.

***Вывод***

Матрица смежности гораздо проще в реализации и сложность проверки наличия ребра составляет всего O(1). Однако, по памяти матрица уступает списку, поскольку требует n\*n памяти. Также при малой компоненте связности матрица получается разреженной.

Алгоритм поиска в ширину реализуется на очереди, в то время как поиск в глубину требует стек и список.