****

|  |  |
| --- | --- |
|  | **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 О ЛАБОРАТОРНОЙ РАБОТЕ №8 ПО ТиСД**

**Графы**

Группа ИУ7-34Б

Название предприятия **НУК ИУ МГТУ имени Н. Э. Баумана**

|  |  |  |
| --- | --- | --- |
| Студент |  | Гареев Георгий Антонович |
| Преподаватель |  | Барышникова Марина Юрьевна |

## Описание условия задачи

Техническое заданиеW

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

Исходные данные

Номер команды меню.

Описание задачи, реализуемой программой

Определяет, является ли связным заданный граф. Строит его графически и выводит матрицы смежности.

Способ обращения в программе

Пользователю выдается меню, в котором он должен выбрать номер функции 0-5:

* 0 – выход из программы
* 1 - Ввести матрицу смежности
* 2 - Вывести матрицу смежности
* 3 - Вывести граф
* 4 - Определить тип графа
* 5 - Вывод меню

Описание возможных аварийных ситуаций и ошибок пользователя

1. Динамическая память не сможет выделится.
2. Попытка вывести\нарисовать пустой граф.

Описание внутренних СД

Для реализации данной задачи создана структура graph, которая используется для хранения графа. size – количество вершин, matrix – указатель на матрицу смежности, transpos\_matrix – указатель на матрицу смежностей в транспонированном виде:

## 

Структуры queue и node используются для реализации очереди, node – узел очереди, queue – хранит указатели на начало и конец очереди.

## 

## Описание алгоритма

Программа позволяет пользователю ввести матрицу смежности. Если для размера матрицы или элемента матрицы были введены некорректные значения – выводится соответствующее сообщение об ошибке.

При помощи меню можно выполнять несколько операций:

1. Определить, является ли граф связным. В случае ошибки во время выполнения (ошибка памяти) – выводится соответствующее сообщение. Иначе выводится сообщение о типе графа.
2. Вывод матрицы смежностей. Если матрица пустая – выводится соответствующее сообщение об ошибке.
3. Вывод графа. Если граф не был создан – выводится соответствующее сообщение об ошибке.

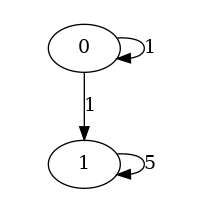
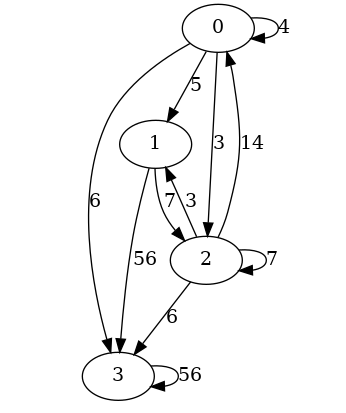
## Тесты

Негативные

1. Неправильный пункт меню
2. Вывод пустого графа и матрицы
3. Определение связанности графа пустого

Позитивные

Вывод корректно введенных графов:

## Информация о реализации

Для определения связности графа был выбран алгоритм поиска в глубину, поскольку нам не важны длины путей, а только факт их существования.

Для определения типа графа использовалась транспонированная матрица смежности, а также очередь – для поиска путей вершины графа, путь к которой уже был найден.

## 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).

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

Графы в памяти могут представляться различным способом. Один из видов представления графов – это матрица смежности B(n\*n); В этой матрице элемент b[i,j]=1, если ребро, связывающее вершины Vi и Vj 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).

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

Перечислим основные операции по работе с графовыми структурами: • 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).

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

Один из основных методов проектирования графовых алгоритмов – это поиск (или обход графа) в глубину (depth first search, DFS).

Указанного недостатка лишен другой метод обхода графа – поиск в ширину (breadth first search, BFS).

1. **Где используются грифовые структуры?**

Графовые структуры используются для моделирования социальных графов (социальных сетей), семантической паутины.

1. **Какие пути в графе Вы знаете?**

Непростой путь, Эйлеров путь, Гамильтонов путь.

1. **Что такое каркасы графа?**

Каркасы графа – это деревья, в которые входят все вершины графа и некоторые его ребра.