

**Министерство образования и науки 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**

### Студент Завойских Евгения Васильевна

*фамилия, имя, отчество*

### Группа ИУ7-33Б

Студент Завойских Е.В.

*подпись, дата фамилия и.о.*

### Руководитель практики Барышникова М.Ю.

*подпись, дата фамилия, и.о.*

### Оценка

*2021 г.*

**Условие 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.

Найти все вершины заданного opграфа, недостижимые из заданной его вершины.

**Внешняя спецификация программы**

1. Входные данные

При вызове программы выводится меню, каждый раз пользователь выбирает один из пунктов.

При выборе нового исходного файла вводится его имя. Вводится номер вершины графа при поиске всех вершин заданного opграфа, недостижимых из заданной его вершины.

1. Выходные данные

По запросу выводятся обычный орграф (не раскрашенный) и орграф с цветными вершинами (зелёный – вершина достижима из заданной вершины, красный - недостижима) – формируются .png файлы.

В случае некорректного ввода команды меню выводится сообщение об ошибке и предоставляется возможность выбрать команду еще раз. В случае некорректного ввода имени исходного файла, ввода номера несуществующей вершины орграфа или чтения некорректного числа из файла также выводится соответствующее сообщение об ошибке. При ошибке выделения памяти выводится соответствующее сообщение об ошибке.

1. Задача, реализуемая программой

Данная программа предназначена для работы с орграфом, и представляет собой приложение со следующими возможными операциями:

1. Ввести имя исходного файла.
2. Вывести орграф.
3. Вывести все вершины заданного орграфа, недостижимые из заданной его вершины.
4. Способ обращения к программе

Программа может быть вызвана из консоли или запущена в любой среде разработки, поддерживающей язык С.

1. Возможные аварийные ситуации и ошибки пользователя

В случае аварийной ситуации программа выводит сообщение о произошедшей ошибке. Ошибка может возникнуть в результате некорректного ввода (не соответствующего формату), при выделении динамической памяти, при отсутствии входных данных при создании орграфа, при открытии исходного файла.

**Внутренние структуры данных**

В программе используются структуры для хранения элемента списка смежностей вершины и для хранения орграфа.

Для элемента:

// структура для определения элемента списка смежностей вершины

struct node

{

    int vertex;  // номер вершины

    struct node \*next;  // указатель на следующую вершину

};

typedef struct node node\_s;

Для орграфа:

// структура для определения орграфа в виде массива из списков смежностей вершин

struct digraph

{

    int size;  // размер массива

    node\_s \*\*arr;  // массив

};

typedef struct digraph digraph\_s;

**Алгоритм**

Пока пользователь не решит завершить программу, выполняется действие, выбранное из меню:

1. Ввести имя исходного файла: с клавиатуры вводится имя файла, если удалось его открыть, читается число вершин в орграфе, выделяется память под массив списков смежностей вершин, далее для каждой вершины под каждую с ней связную вершину выделяется память и новый элемент добавляется в список смежностей текущей вершины
2. Вывести орграф: в файл записывается орграф на языке DOT, по нему формируется .png файл с изображением орграфа
3. Вывести все вершины заданного орграфа, недостижимые из заданной его вершины: для введенной вершины производится поиск в глубину, каждая посещенная вершина отмечается в массиве посещенных вершин, все не посещенные выводятся на экран, в файл записывается орграф на языке DOT, по нему формируется .png файл с изображением орграфа, где посещенные вершины закрашены зеленым цветом, а не посещенные - красным

**Тесты**

Негативы:

|  |  |  |
| --- | --- | --- |
| Входные данные | Вывод | Что проверяется |
| Возможные действия (введите номер нужной команды):  0. Завершить работу  1. Ввести имя исходного файла  2. Вывести орграф  3. Вывести все вершины заданного opграфа, недостижимые из заданной его вершины  6 | Нет команды с введенным номером | Неверно введена команда меню |
| 2  Орграф еще не создан | Орграф не создан | Попытка вывести орграф до создания |
| 1  2.txt  Файл пустой | Ошибка ввода | Пустой ввод |
| C:\msys64\home\zavoy\tisd_labs\lab_7\digraph.png  3  Введите номер вершины: 11 | Ошибка ввода | Попытка вывести недостижимые вершины для несуществующей вершины |
| 1  3.txt  В файле:  3  1 0 0 | Ошибка ввода | Таблица смежности введена не целиком |
| 1  4.txt  В файле:  2  1 2  0 0 | Ошибка ввода | В таблице смежности из файла некорректное значение |
| 1  10.txt | Ошибка открытия файла | Введено имя несуществующего файла |

Позитивы:

|  |  |  |
| --- | --- | --- |
| Входные данные | Вывод | Что проверяется |
| 1  1.txt  2 | C:\msys64\home\zavoy\tisd_labs\lab_7\digraph.png | Вывод созданного орграфа |
| 3  Введите номер вершины: 2 | Все недостижимые вершины:  1 3 4 5 6 7 8  C:\msys64\home\zavoy\tisd_labs\lab_7\visited.png | Из вершины нельзя достичь ни одной другой вершины |
| 3  Введите номер вершины: 4 | Все недостижимые вершины:  C:\msys64\home\zavoy\tisd_labs\lab_7\visited.png | Из вершины достижимы все вершины |
| 3  Введите номер вершины: 1 | Все недостижимые вершины:  4 5 6  C:\msys64\home\zavoy\tisd_labs\lab_7\visited.png | Из вершины достижимы не все другие вершины |

**Вывод**

Для поиска недостижимых вершин в графе из данной вершины подходят алгоритмы поиска в глубину, в ширину и вычисление матрицы достижимости. Временная сложность первых двух алгоритмов равна 𝑂(|V|+|Е|), для матрицы достижимости - 𝑂(|V|^3). Я выбрала поиск в глубину, так как он имеет наименьшую временную сложность и прост в реализации.

Для хранения орграфа использовался массив списков смежностей, так как он позволяет при поиске в глубину просматривать только смежные с текущей вершины. При использовании матрицы смежностей нам пришлось бы дополнительно к поиску проверять каждую вершину на смежность с текущей. Выбранная структура позволяет экономить время и память по сравнению с матрицей смежности при небольшом числе ребер в орграфе, если в графе число ребер близко к |V|^2, есть смысл использовать матрицу смежности для экономии памяти.

Такая программа может применяться при решении следующей задачи: имеется компьютерная сеть, состоящая из центров хранения и обработки информации. Некоторые пары центров соединены каналами. Обмен информацией между любыми двумя центрами осуществляется в одну сторону либо непосредственно по соединяющему их каналу, если он есть, либо через другие каналы и центры. Такой сети можно сопоставить граф: вершины – центры, ребра – каналы сети. Затем, используя написанную программу, можно проверить, из какого центра можно передать информацию в любой другой центр.

**Ответы на 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). Если пары Е (ребра) имеют направление, то граф называется ориентированным (орграф), если иначе - неориентированный (неорграф).

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

С помощью матрицы смежности или списков смежности. В матрице смежности элемент b[i,j]=1, если ребро, связывающее вершины Vi и Vj существует и b[i,j]=0, если ребра нет. Список смежностей содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с этой вершиной.

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

Обход вершин, поиск различных путей между вершинами, исключение и включение вершин.

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

Обход в ширину и обход в глубину.

1. **Где используются графовые структуры?**

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи, необязательно иерархически.

1. **Какие пути в графе Вы знаете?**

Эйлеров путь, непростой путь, гамильтонов путь. Произвольный путь в графе, проходящий через каждое ребро графа точно один раз, называется эйлеровым путем. При этом, если по некоторым вершинам путь проходит неоднократно, то он является непростым. Путь в графе, проходящий в точности один раз через каждую вершину графа (а не каждое ребро), называется гамильтоновым и существуют не для каждого графа, как и эйлеров путь.

1. **Что такое каркасы графа?**

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