



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 №08 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

Графы

Студент **Батуев А.Г.**

Группа **ИУ7-36Б**

Вариант **1**

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

Студент _____ **Батуев А.Г.**

Преподаватель _____ **Никульшина Т.А.**

2024 г.

Условие задачи

Цель работы – реализовать алгоритмы обработки графовых структур. Обработать графовую структуру в соответствии с указанным вариантом задания.

Задание по варианту.

Найти все вершины заданного орграфа, недостижимые из заданной его вершины

Техническое задание

На вход программа получает:

- опция из меню (цифра от 0 до 3)
- файл с данными
- данные от пользователя

Программа выводит:

- граф в формате .dot
- матрица связности
- недостижимые вершины

Программа должна реализовывать создание, хранение, вывод графа, а также операции экспорта и поиска вершин.

Обращение к программе осуществляется по указанию названия программы (./app.exe).

Необходимо внимательно **обработать возможные аварийные ситуации**, которые могут включать:

- неправильный ввод данных
- переполнение памяти

Описание внутренних структур данных

graph_t — это структура, представляющая граф. Граф содержит:

- *vertices* - целое число, представляющее количество вершин в графе.
- *data* - указатель на целые числа, представляющий матрицу смежности графа (если *dij* элемент не равен нулю значит между вершинами есть дорога).

```
typedef struct
{
    int vertices;
    int** data;
} graph_t;
```

Основные функции программы

Программа реализует следующие основные функции:

- `void action(int option, tree_t *tree hash_chain *chain, hash_open *open);`
Функция меню:
 - Сохранить в dot
 - Матрица смежности
 - Недостижимые вершины
 - Выход
- `void read_graph_from_file(graph_t* graph, char* filename);` Эта функция предназначена для чтения данных графа из файла
- `void save_graph_to_dot(graph_t *graph, char *filename);` Сохранить граф в формате dot
- `void find_unreachable(graph_t *graph, int start);` Поиск недостижимых вершин
- `void print_graph(graph_t* graph);` Вывод матрицы смежности

Описание алгоритма поиска недостижимой вершины

Алгоритм использует поиск в глубину (DFS) для определения всех вершин, достижимых из заданной начальной вершины. Затем он сравнивает множество всех вершин с множеством достижимых вершин, чтобы найти вершины, которые не были посещены, то есть являются недостижимыми.

Набор тестов

Тестовый сценарий	Входные данные	Ожидаемый результат
Поиск недостижимых вершин	Структура graph_t с 5 вершинами, ребра: 0->1, 1->2, 3->4 начальная вершина 0.	Вывод в консоль: "Недостижимые вершины из данной: 3 4\n"
Поиск недостижимых из изолир. вершины	Структура graph_t с 5 вершинами, ребра: 0->1, 1->2, 3->4, начальная вершина 3.	Вывод в консоль: "Недостижимые вершины из данной: 0 1 2\n"
Все вершины достижимы	Структура graph_t с 3 вершинами, ребра: 0->1, 1->2, 2->0, начальная вершина 0	Вывод в консоль: "Недостижимые вершины из данной: отсутствуют\n"
Чтение из несуществующего файла	nonexistent.txt	Сообщение об ошибке "Ошибка открытия файла" выводится на экран.

Контрольные вопросы

1. Что такое граф

Граф – это конечное множество вершин и ребер, соединяющих их.

2. Как представляются графы в памяти?

Графы в памяти могут представляться различным способом. Виды представления графов:

- матрица смежности; Представляется в виде матрицы, где значение 0 соответствует отсутствию дороги, а не нулю существованию.
- список смежностей; Для каждой вершины содержит список тех вершин, которые непосредственно связаны с этой вершиной.

3. Какие операции возможны над графами?

Добавление, удаление вершин/дорог, поиск кратчайшего/мин стоимости пути, обход.

4. Какие способы обхода графов существуют?

Поиск в глубину (DFS), поиск в ширину (BFS), двунаправленный поиск, итеративный поиск в глубину.

5. Где используются графовые структуры?

- Компьютерные сети: представление узлов сети (серверы, маршрутизаторы) и соединений между ними.
- Нейронные сети: в некоторых случаях, например, графовые нейронные сети, графовая структура используется для анализа данных.
- Карты и навигации
- Нейроны мозга

6. Какие пути в графе Вы знаете?

Путь от А до Б, кратчайший путь, путь по меньшей стоимости, гамильтонов путь.

7. Что такое каркасы графа?

Каркасом, или остовным деревом для этого графа называется связный подграф этого графа, содержащий все вершины графа и не имеющий циклов. Количество ребер в каркасе связного графа всегда на единицу меньше количества вершин графа.

Существует несколько алгоритмов для нахождения каркасов графа, наиболее известные из них:

- алгоритм Прима. Начинает с произвольной вершины и постепенно добавляет ребра с наименьшим весом, пока не будет построен каркас.

- алгоритм Краскала. Сортирует все ребра по возрастанию веса и добавляет их в каркас, если они не создают циклов.

Заключение

Граф – структура, которая используется, когда один элемент связан с другими. Он помогает проанализировать связи между объектами.