



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 №8 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ

“Графы”

Студент **Зернов Георгий Павлович**

Группа **ИУ7-34Б**

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

Студент _____ **Зернов Г.П.**

Преподаватель _____ **Никульшина Т.А.**

Оглавление

Условие задачи.....	3
Техническое задание.....	4
Реализуемая в программе задача.....	4
Входные данные.....	4
Выходные данные.....	4
Способ обращения к программе.....	5
Возможные аварийные ситуации или ошибки пользователя.....	5
Внутренние структуры и типы данных.....	5
Алгоритм программы.....	6
Функции программы.....	7
Тесты.....	7
Анализ эффективности.....	9
Пример реальной задачи.....	10
Вывод.....	10
Контрольные вопросы:.....	11

Условие задачи

Найти самый длинный простой путь в графе. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Предложить вариант реальной задачи, для решения которой можно использовать разработанную программу. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

Техническое задание

Реализуемая в программе задача

Цель работы: реализовать алгоритмы обработки графовых структур: поиск различных путей, проверка связности, построение остовых деревьев минимальной стоимости.

Входные данные

В случае выбора опции в меню целое число, соответствующее одной из указанных опций:

- 1 – Загрузить граф из файла
- 2 – Найти самый длинный простой путь в файле
- 3 – Вывести граф в файл
- 4 – Выход

В случае загрузки графа из файла:

- Имя файла – строка.

В случае вывода графа в файл:

- Имя файла – строка.

Выходные данные

1. В случае поиска самого длинного простого пути – самый длинный простой путь, выведенный на экран.
2. Вывести граф в файл – файл содержащий граф в dot формате.

Способ обращения к программе

Чтобы обратиться к программе, пользователю необходимо запустить файл app.exe.

Возможные аварийные ситуации или ошибки пользователя

- Введено невалидное значение опции – вывод на экран информирующего сообщения о неверном вводе и предоставление пользователю возможности повторно ввести значение.
- Введено невалидное имя файла - вывод на экран информирующего сообщения о неверном вводе и предоставление пользователю возможности повторно ввести имя файла.
- Ввод из файла, который невозможно открыть на чтение или содержит невалидную информацию (в том числе 0-граф) – вывод информирующего сообщения.
- Вывод в файл, который невозможно открыть на запись – вывод информирующего сообщения.
- Нехватка памяти при её выделении – соответствующее сообщение об ошибке и остановка текущего процесса с возвратом программы в состояние до его начала.

Внутренние структуры и типы данных

Структура, представляющее тип данных “граф”.

- `size_t vertex_num` – число вершин в графе.
- `int **adj_matrix` – матрица смежности графа.

Листинг структуры:

```
typedef struct
{
    size_t vertex_num;
    int **adj_matrix;
} graph_t;
```

Структура, представляющее тип данных “путь”.

- size_t path_len – длина текущего пути.
- int *path – массив номеров посещённых вершин.

Листинг структуры:

```
typedef struct
{
    size_t path_len;
    int *path;
} path_t;
```

Алгоритм программы

- Общий алгоритм программы:

1. Запуск основного цикла программы.
2. Вывод меню.
3. Запрос у пользователя опции для выполнения.
4. Выполнение выбранной опции.
5. Переход к новой итерации цикла или завершение основного цикла при соответствующем выборе пользователя.

- Алгоритм поиска самого длинного пути в графе:

1. Для каждой из вершин запускается алгоритм поиска в глубину.
2. Поиск сохраняет информацию о самом длинном пути, текущем пути и посещённых вершинах, если самый длинный путь короче текущего –


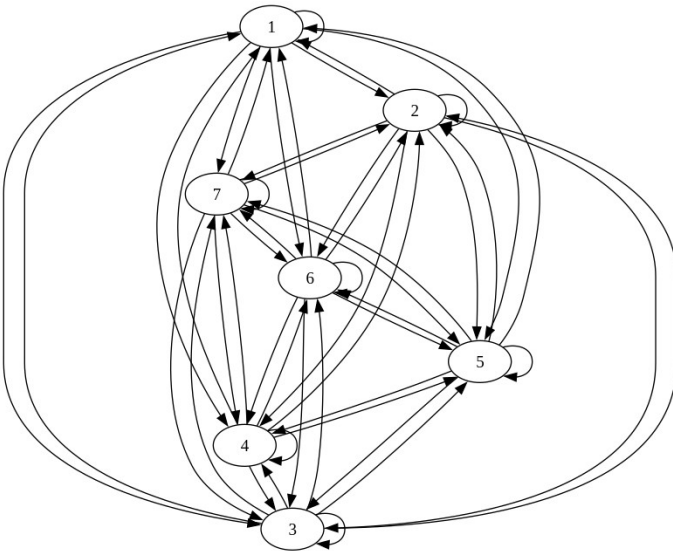
обновляет его, если самый длинный путь по длине равен числу вершин графа, то алгоритм завершается.

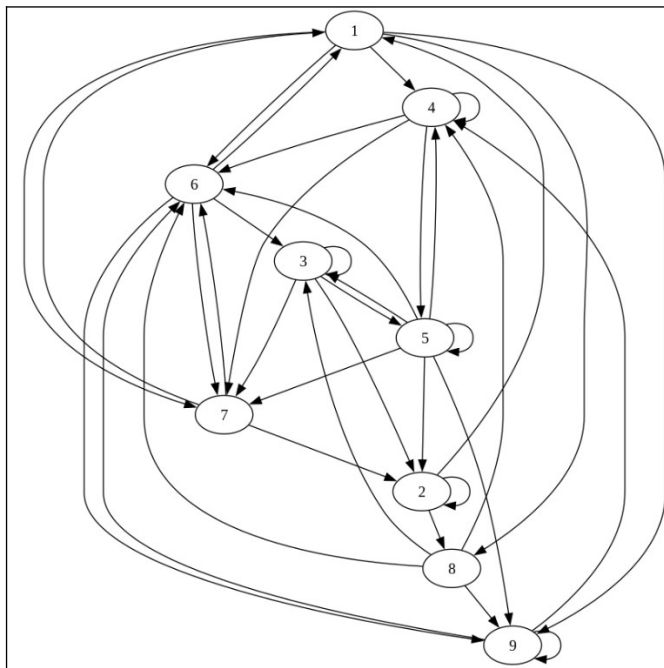
Функции программы

- `tree_node_t* create_node(char *data)` – создаёт узел по содержимому, возвращает указатель на узел.
- `void destroy_graph(graph_t *graph)` – освобождает граф.
- `int import_graph(FILE *file, graph_t *dst)` – импортирует граф из файла, возвращает код ошибки.
- `void export_graph_to_dot(FILE *file, graph_t *graph)` – экспортирует граф в файл, в dot формат, возвращает код ошибки.
- `void find_max_simple_path(graph_t *graph, path_t *max_path)` – ищет самый длинный простой путь в графе.
- `void print_path(path_t *path)` – выводит на экран самый длинный путь.
- `void print_menu(void)` – выводит меню.
- `int main(void)` – выполняет основной алгоритм программы.

Тесты

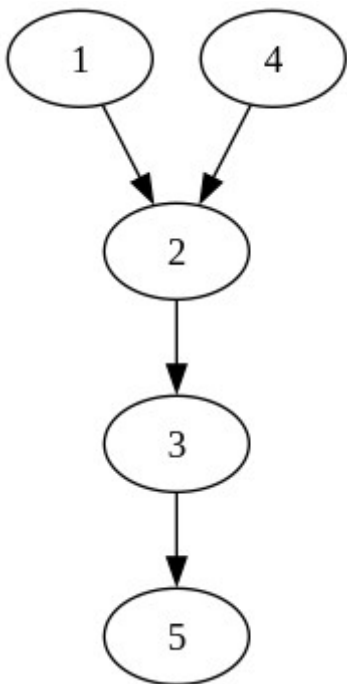
Тестовый случай	Ожидаемый результат
Введена невалидная опция	Соответствующее сообщение об ошибке, возможность снова ввести требуемое значение.
Введено невалидное имя файла ввода	Соответствующее сообщение об ошибке, возможность снова ввести имя файла.
Чтение из файла ввода невозможно	Соответствующее сообщение об ошибке.
Введено невалидное имя файла вывода	Соответствующее сообщение об ошибке,

	возможность снова ввести имя файла.
Запись в файл вывода невозможна	Соответствующее сообщение об ошибке.
Открытие файла с графом и его считывание	Успешное считывание файла
Вывод графа в файл	Успешный вывод графа в dot формате
Поиск самого длинного простого пути Граф: 	Самый длинный простой путь: 1
Поиск самого длинного простого пути Граф: 	Самый длинный простой путь: 1-2-3-4-5-6-7-8-9-10
Поиск самого длинного простого пути Граф:	Самый длинный простой путь: 1-4-6-3-5-7-2-8-9



Поиск самого длинного простого пути

Граф:



Самый длинный простой путь:

1-2-3-5

Анализ эффективности

Поиск самого длинного простого пути в графе является NP-hard задачей, соответственно для её решения применим полный перебор. Сложность перебора $O(n^2)$.

Так как плотность графа неизвестна, граф ориентированный, то можно считать, что способ хранения графа через матрицу смежности является достаточно эффективным.

Пример реальной задачи

Задача: требуется выполнить наибольшее число, зависящих друг от друга задач.

Вывод

Поиск самого длинного пути в произвольном ориентированном графе является NP-hard задачей, из чего следует, что он находится полным перебором. Для полного перебора был использован DFS, вызываемый от каждой вершины, что является достаточно эффективным решением по времени. По памяти решение не является эффективным, так как является рекурсивным. Эффективным по памяти способом хранения является матрица смежности, так как граф ориентированный, а его плотность неизвестна, соответственно в среднем матрица смежности будет не менее эффективна, чем список смежности и матрица инцидентности.

Контрольные вопросы:

Что такое граф?

Граф — это структура данных, состоящая из множества вершин (узлов) и рёбер (связей между вершинами).

Как представляются графы в памяти?

- Список смежности: Для каждой вершины хранится список соседних вершин.
- Матрица смежности: Двумерный массив, где элемент $[i][j]$ равен весу ребра между вершинами i и j или 1 – если связь есть, иначе – 0.
- Матрица инцидентности: двумерная матрица, где строки — вершины, а столбцы — рёбра, 1 – если вершина инцидентна ребру, иначе – 0.

Какие операции возможны над графами?

- Добавление/удаление вершин.
- Добавление/удаление рёбер.
- Поиск пути между вершинами.
- Проверка связности графа.
- Поиск кратчайшего пути.
- Поиск минимального остовного дерева (каркаса).
- Обход графа.

Какие способы обхода графов существуют?

- Поиск в глубину (DFS, Depth-First Search)
- Поиск в ширину (BFS, Breadth-First Search)

Где используются графовые структуры?

В случаях, когда требуется представить попарные связи между объектами, например:

- Компьютерные сети
- Навигация (построение путей)
- Алгоритмы поиска
- Базы данных (зависимости между записями)
- и.т.д

Какие пути в графе Вы знаете?

- Простой путь – путь, в котором все вершины уникальны.
- Эйлеров путь – проходит через каждое ребро графа ровно один раз.
- Эйлеров цикл – Эйлеров путь, который начинается и заканчивается в одной вершине.
- Гамильтонов путь – проходит через каждую вершину ровно один раз.
- Гамильтонов цикл – Гамильтонов путь с возвращением в начальную вершину.
- Кратчайший путь – путь с минимальной суммарной стоимостью рёбер.

Что такое каркасы графа?

Каркас графа (минимальное остовное дерево) — это подграф, соединяющий все вершины исходного графа минимальным числом рёбер без образования циклов.