



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 Лабораторная работа №8 «Графы»

Группа ИУ7-34Б

Дисциплина Типы и структуры данных

Вариант 14

Студент

Козлитин Максим Александрович

Преподаватель

Силантьева Александра
Васильевна

2022г.

Цель работы

Реализовать алгоритмы обработки графовых структур: поиск путей минимальной стоимости.

Условие задачи

Найти все вершины графа, к которым от заданной вершины можно добраться по пути не длиннее A.

Описание исходных данных

- Выбор действия:
Целое число от 0 до 3.
- Вес ребра
Целое число от 0 до 1000
- Вершина
Целое число от 0 до указанонного максимального числа вершин

Способ обращения к программе

Работа с программой осуществляется с помощью консоли.

Программа запускается с помощью команды: `./app.exe`

Далее пользователь выполняет взаимодействие с помощью меню. Меню выводится в начале и, далее, каждые 3 введенные команды.

Просмотр графиков выполняется в два действия:

1. Конвертация графика в изображение «dot -Tpng graph.gv > graph.png»
2. Просмотр изображения

Описания возможных аварийных ситуаций и ошибок пользователя

1. Ввод не совпадающий с форматом входных данных (описано в Описание исходных данных).
2. Ввод целых чисел превышающих максимальное допустимое значение типа данных int.

Описание внутренних структур данных

Граф хранится в виде матрицы смежности.

Матрица смежности:

```
typedef struct
{
    size_t size; // количество вершин
    int **data; // содержимое матрицы
} adjacency_matrix_t; // матрица смежности графа
```

size_t используется для задания размеров наших массивов, так как является особым макросом, значение которого позволяет адресовать массив любой теоретически возможной длины для данной машины.

int - целочисленный тип данных.

Описание алгоритма

Поиск кратчайших путей (алгоритм Дейкстры) -

adjacency_matrix_shortest_paths:

1. Отмечаем что кратчайший путь до нашей вершины = 0
2. Пока все возможные расстояния до вершин не подсчитаны, выбираем вершины до которой минимальное расстояние
3. Проводим релаксацию от этой вершины. Т.е. если расстояние от выбранной вершины меньше, чем расстояние до релаксируемой вершины, то мы уменьшаем его.

Оценка эффективности

N - количество вершин

Поиск минимума $\sim O(N)$

Релаксация одной вершины $\sim O(1)$

Релаксация всех вершин $\sim O(N) * O(1) \sim O(N)$

Общее $\sim O(N) * O(N) \sim O(N^2)$

Оценка использования памяти

так как граф хранится в виде матрицы смежности, то требуется хранить матрицу размером $N*N$

Каждый элемент матрицы хранит вес ребра - int

Размер - $N*N*\text{sizeof}(\text{int})$

Вариант реальной задачи

Недавно мэр г. Москва - Сергей Семёнович Собянин, захотел узнать до каких парков города Москвы можно добраться на метро меньше чем за 20 минут. На карте метро отмечено за сколько состав преодолевает участок от одной станции до следующей. Требуется представить в графическом формате ответ на данную задачу.

Вывод

В данной лабораторной работе был использован алгоритм Дейкстры для поиска кратчайших путей, так как нам нужно находить расстояние от одной вершины до всех остальных, при этом граф является взвешенным.

Использование алгоритма Флойда-Уоршалла было бы избыточным так как нам не требуется расстояние от каждой вершины до каждой.

Использование BFS не позволяет решить данную задачу, так как граф - взвешенный.

Не использование алгоритма Форда-Беллмана обусловлено масштабируемостью алгоритм Дейкстры для разреженных графов. Так как исходный граф не допускает отрицательных весов, то использованием алгоритма Форда-Беллмана можно пренебречь.

Для хранения графа был использована матрица смежности, это обусловлено скоростью работы. При релаксации требуется часто получать доступ к элементу, при использовании списка смежности, потребовалось бы больше накладных расходов на данную операцию.

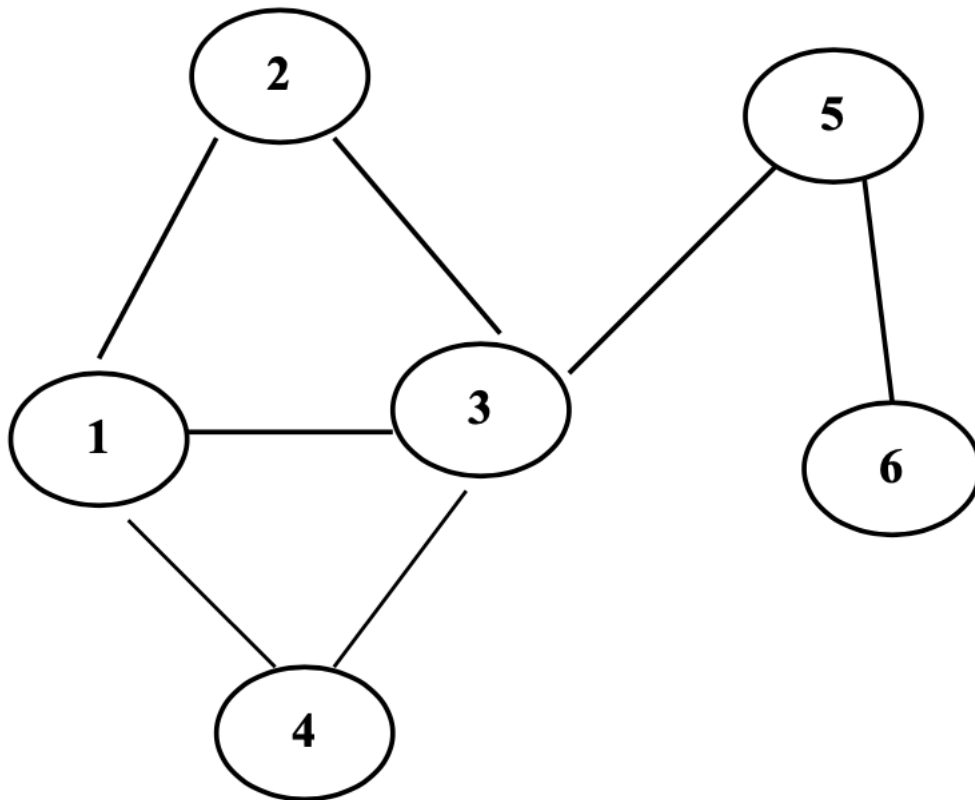
При этом приходится жертвовать памятью, но в условиях ограничения на количество вершин в графе, а также доступность оперативной памяти в

современном мире, можно отдать предпочтения матрице смежности в пользу скорости.

Контрольные вопросы

1. Что такое граф?

это конечное множество вершин и ребер, соединяющих их.



2. Как представляются графы в памяти?

В виде матрицы смежности - матрица где ячейка $matrix[i][j]$ содержит вес ребра из вершины i в вершину j

В виде списка смежности - массива списков, где каждый элемент массива является списком ребер данной вершины.

3. Какие операции возможны над графами?

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 (DFS) - спускаемся от каждой вершины на максимальную глубину.
Обход в ширину (BFS) - от текущей вершины расширяем обход по новым соседям.

5. Где используются графовые структуры?
Графы можно масштабировать для использования во многих задачах.
Одним из применений является имитация дорог или путей.
Например:
Поиск суммарных дорог между городами, чтобы длина была минимальной.
6. Какие пути в графе вы знаете?
Простой путь - путь, в котором все вершины попарно различны (каждая вершина присутствует единожды)
Эйлеров путь - путь, проходящий через все вершины графа
Гамильтонов путь - простой путь, проходящий через все вершины графа
Кратчайший путь - путь наименьшей стоимости из одной вершины в другую
7. Что такое каркасы графа?
Каркас или остовное дерево - связный подграф данного графа, содержащий все вершины и не содержащий циклов.

