

Лабораторная работа № 9 по курсу дискретного анализа: Графы

Выполнил студент группы М8О-307Б-20 МАИ *Чекменев Вячеслав*.

Условие

1. Разработать программу на языке C или C++, реализующую указанный алгоритм. Формат входных и выходных данных описан в варианте задания.
2. **Вариант 4: Поиск кратчайшего пути между парой вершин алгоритмом Дейкстры.** Задан взвешенный неориентированный граф, состоящий из n вершин и m рёбер. Вершины пронумерованы целыми числами от 1 до n . Необходимо найти длину кратчайшего пути из вершины с номером *start* в вершину с номером *finish* при помощи алгоритма Дейкстры. Длина пути равна сумме весов ребер на этом пути. Граф не содержит петель и кратных рёбер.

Метод решения

Алгоритм Дейкстры является одним из самых распространённых способов найти кратчайший путь между двумя вершинами в графе. Его недостатком является то, что он не работает в графах с рёбрами с отрицательным весом. Его преимущество заключается в том, что он ищет кратчайшие пути от заданной вершины до вообще всех остальных вершин (хотя в рамках поставленной задачи это будет лишним).

Сначала каждой вершине графа сопоставим расстояние от неё до начальной вершины, у начальной вершины это будет очевидно 0, у всех остальных бесконечность. Также создадим вектор `bool`, где будем хранить информацию о том, посетили ли мы вершину.

Создадим множество, в котором будем хранить пары (расстояние до вершины, вершина), множество будет автоматически отсортировано по возрастанию. Вставим в множество нашу стартовую вершину и расстояние до нее.

Будем идти циклом по множеству пока оно не пусто. На каждой итерации будем производить релаксацию ребер исходящих из вершины в начале множества. В итоге мы получим массив с минимальными путями до всех вершин.

Описание программы

Программа состоит из одного файла.

Дневник отладки

1. неправильно прочитал задание, думал, что граф ориентированный
2. Написал сначала алгоритм за $N^2 + M$, *pqqueue*,

Тест производительности

Кол-во вершин	кол-во ребер	Время (в мс)
n = 10	m = 15	0m0,015s
n = 100	m = 150	0m0,017s
n = 1000	m = 1500	0m0,038s
n = 10000	m = 15000	0m0,101s
n = 100000	m = 150000	0m1,180s
n = 1000000	m = 1500000	0m12,990s

Сложность $O(m \cdot \log(n))$

Недочёты

Можно ускорить программу если использовать рqueue.

Выводы

Проделав лабораторную работу, реализовал вариацию алгоритма Дейкстры, имеющего широкое прикладное применение в сетях и маршрутах транспорта. Алгоритм оказался не настолько сложным, однако довольно быстрым, ообенно на разреженных графах.