

Лабораторная работа № 9 по курсу дискретного анализа: Графы

Выполнил студент группы М8О-308Б-20 МАИ *Попов Матвей*.

Условие

1. Разработать программу на языке С или С++, реализующую указанный алгоритм. Формат входных и выходных данных описан в варианте задания. Первый тест в проверяющей системе совпадает с примером.
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).

Первым делом сопоставим каждой вершине графа некое число — то самое наименьшее расстояние от неё до начальной вершины, у начальной вершины это будет очевидно 0, у всех остальных бесконечность, то есть достаточно большое число, затем для каждой вершины, связанной с начальной ровно 1 ребром проверим, является ли сопоставленное ей в начале число больше, чем сумма такого же числа той вершины, из которой мы пришли и веса ребра, по которому пришли. Если является, заменяем число на эту самую сумму. Как только обошли всех соседей вершины, помечаем вершину как пройденную, это значит, что минимальное расстояние до этой вершины найдено и изменять его больше не надо. То же самое делаем со всеми остальными вершинами, пользуясь поиском в глубину.

Описание программы

Программа состоит из одного файла.

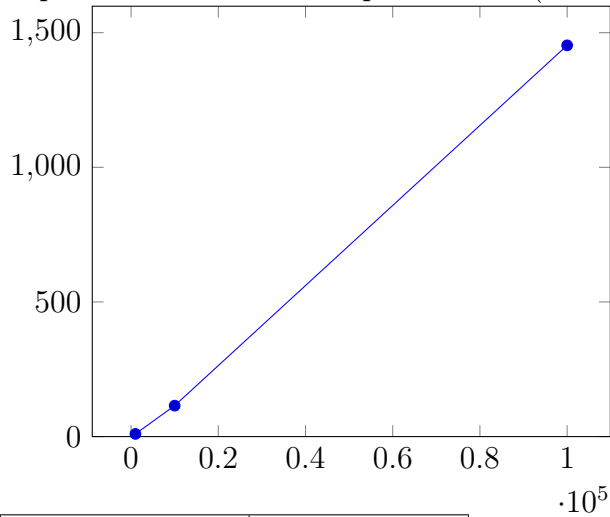
Дневник отладки

1. Наученный горьким опытом предыдущих задач, я заменил тип `int32` на `int64`.

2. Программа переписана с нуля, заменены некоторые структуры данных.
3. Вместо структуры TEdge для рёбер было принято решение использовать `std::pair`. Это оказалось удобнее.

Тест производительности

Ниже приведен тест времени работы алгоритма. По оси X — количество вершин в графе (количество рёбер на каждом тесте в e раз больше количества вершин), по оси Y — время выполнения алгоритма в мс (меньше — лучше).



Кол-во вершин	Время (в мс)
1000	10
10000	115
100000	1453

Мы не получили ожидаемой сложности $O(n^2)$, так как наш алгоритм завершается сразу, как только конечная вершина оказалась помеченной. Благодаря этому усхоернию сложность стремится к линейной.

Недочёты

Эта лабораторная работа стала первой и, к сожалению, последней, где я не могу выявить очевидных недочётов.

Выводы

Проделав лабораторную работу, реализовал вариацию алгоритма Дейкстры, имеющего широкое прикладное применение в построении маршрутов и не только.