

Лабораторная работа № 9 по курсу дискретного анализа: Графы

Выполнил студент группы М8О-308Б-21 *Армишев Кирилл*.

Вариант 4: Поиск кратчайшего пути между парой вершин алгоритмом Дейкстры

Задан взвешенный неориентированный граф, состоящий из n вершин и m ребер. 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 ребер.

Метод решения

Алгоритм Дейкстры — алгоритм на графах, изобретённый нидерландским учёным Эдсгером Дейкстрой в 1959 году. Находит кратчайшие пути от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании.

Каждой вершине из V сопоставим метку — минимальное известное расстояние от этой вершины до a . Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

Инициализация

Метка самой вершины a полагается равной 0, метки остальных вершин — бесконечности. Это отражает то, что расстояния от a до других вершин пока неизвестны. Все вершины графа помечаются как непосещённые.

Шаг алгоритма

Если все вершины посещены, алгоритм завершается. В противном случае, из ещё не посещённых вершин выбирается вершина u , имеющая минимальную метку. Мы рассматриваем всевозможные маршруты, в которых u является предпоследним пунктом. Вершины, в которые ведут рёбра из u , назовём соседями этой вершины. Для каждого соседа вершины u , кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки u и длины ребра, соединяющего u с этим соседом.

Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину u как

посещённую и повторим шаг алгоритма.

Сложность алгоритма: $O((V + E) * \log(V))$ (В связи с использованием приоритетной очереди)

Описание программы

Реализация довольно простая, состоит из следующих шагов:

1. Создание структуры `wedge` представляет собой ребро графа со следующими полями:
u: вершина начала ребра, v: вершина конца ребра, w: вес (длина) ребра.
2. `wgraph` представляет собой взвешенный граф, представленный в виде вектора векторов ребер.
3. Основная функция `dks` принимает начальную вершину `start` и взвешенный граф `g` и возвращает вектор кратчайших расстояний от `start` до всех остальных вершин.
4. В `int main()` мы вызываем функцию `dks` для поиска кратчайших расстояний от `start` до всех вершин. Выводим результат: если расстояние до конечной вершины не равно 0 и не равно бесконечности, выводим расстояние, иначе выводим "No solution".

Дневник отладки

1. 16 ноя 2023, 23:35:34 WA на 8 тесте.

Причина: В конце в случае `d[finish]==INF` выводил неправильный ответ, вместо вывода "No solution"

Тест производительности

Тесты представляют из себя файлы, в которых сгенерировано различное число ребер.

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

Выводы

Выполнив данную лабораторную работу, я больше узнал о графах и алгоритмах работы с ними. Познакомился с алгоритмом Дейкстры. Алгоритм Дейкстры может найти кратчайший путь между вершинами `s` и `t` в графе, только если существует хотя бы один

путь между этими вершинами. Если это условие не выполняется, то алгоритм отрабатывает корректно, вернув значение "бесконечность" для пары несвязанных вершин. Стоит отметить, что алгоритм не работает, когда в графе существуют отрицательные веса. Для этого существует подход динамического программирования — алгоритм Беллмана – Форда. Несмотря на это, алгоритм Дейкстры является простым и мощным алгоритмом для решения прикладных задач.