

Лабораторная работа № 9 по курсу дискретного анализа: графы

Выполнил студент группы 80-208 МАИ *Сизонов Артём*.

Условие

Кратко описывается задача:

1. Разработать программу на языке C или C++, реализующую указанный алгоритм. Формат входных и выходных данных описан в варианте задания. Первый тест в проверяющей системе совпадает с примером.
2. Вариант задания: **5. Поиск кратчайшего пути между парой вершин алгоритмом Беллмана-Форда.** Задан взвешенный ориентированный 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 Беллмана-Форда. Длина пути равна сумме весов ребер на этом пути. Обратите внимание, что в данном варианте веса ребер могут быть отрицательными, поскольку алгоритм умеет с ними работать. Граф не содержит петель, кратных ребер и циклов отрицательного веса.

Метод решения

Решение задачи заключается в последовательном применении процедуры релаксации для каждого ребра. Описание процедуры релаксации: пусть имеется массив расстояний *dist*. Элемент массива *dist[i]* хранит расстояние от стартовой вершины до вершины с номером i . Пусть в данный момент мы обрабатываем ребро с началом в вершине a , концом в вершине b и весом w . Тогда, если текущее расстояние до вершины b больше, чем расстояние до вершины a плюс вес обрабатываемого ребра, то необходимо заменить *dist[b]* на *dist[a] + w*. В данном алгоритме утверждается, что достаточно $n - 1$ фазы алгоритма, на каждой из которых происходит процедура релаксации для всех ребер, чтобы посчитать длины всех кратчайших путей в графе от заданной начальной вершины. Также в данном алгоритме происходит следующая оптимизация. Если на какой-то фазе не происходит никаких изменений в массиве расстояний, то и на следующем шаге также не произойдет никаких изменений. Следовательно, выполнение алгоритма можно остановить.

Описание программы

Программа написана одним файлом.

struct TEdge – предназначенная для описания ребра графа структура.

Дневник отладки

Посылка 1: *Status:* Wrong answer at test 11.t. Проблема: для вычисления промежуточных значений использовался тип *int*, а не *long long*.

Посылка 2: *Status:* Time limit exceeded at test 17.t. Проблема: в алгоритме не использовалась оптимизация.

Посылка 3: *Status:* ОК.

Тест производительности

Количество ребер и вершин с каждым тестом умножается на 5.

```
bsb@dell:~/Рабочий стол/Информатика/Дискретный анализ/Лабораторные/LabN9$ time cat test1 | ./b.out > ans
real    0m0.010s
user    0m0.000s
sys     0m0.004s
bsb@dell:~/Рабочий стол/Информатика/Дискретный анализ/Лабораторные/LabN9$ time cat test2 | ./b.out > ans
real    0m0.015s
user    0m0.008s
sys     0m0.000s
bsb@dell:~/Рабочий стол/Информатика/Дискретный анализ/Лабораторные/LabN9$ time cat test3 | ./b.out > ans
real    0m0.119s
user    0m0.104s
sys     0m0.000s
bsb@dell:~/Рабочий стол/Информатика/Дискретный анализ/Лабораторные/LabN9$ time cat test4 | ./b.out > ans
real    0m2.620s
user    0m2.552s
sys     0m0.000s
```

Недочёты

Выявленных недочётов нет.

Выводы

Графы имеют огромное использование в программировании. Одной из важнейших задач является поиск кратчайшего пути во взвешенном графе. Реализованный алгоритм Беллмана-Форда решает данную задачу. Его временная сложность: $O(n \cdot m)$, где n – количество вершин в графе, а m – количество ребер. Одно из преимуществ данного алгоритма заключается в том, что он допускает рёбра с отрицательным весом. Для корректной работы алгоритма заданный граф не должен содержать циклов отрицательного веса. С помощью данного алгоритма также можно определить, существует ли в графе такой цикл.