

# **Лабораторная работа №9 по курсу дискретного анализа: Графы**

Выполнил студент группы М8О-307Б-20 Мерц Савелий Павлович

## **Условие**

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.

## **Вариант:**

Задан взвешенный ориентированный граф, состоящий из  $n$  вершин и  $m$  ребер. Вершины пронумерованы целыми числами от 1 до  $n$ . Необходимо найти длины кратчайших путей между всеми парами вершин при помощи алгоритма Джонсона.

## **Метод решения:**

В алгоритме Джонсона используются алгоритмы Дейкстры и Беллмана-Форда, поэтому сначала я разработал их.

Граф я храню в списке смежности, реализованный с помощью `vector<unordered_map<int, long long>>` где ключ - это номер вершины, куда ведёт ребро, а значение - это длина ребра.

### **Алгоритм Дейкстры:**

Необходим контейнер для хранения ответа, контейнер для того, чтобы отмечать в каких вершинах уже побывал, и как-то находить или хранить кандидата для посещения. Для хранения результата я использовал `vector<long long>`, для отметок `vector<bool>`, для кандидатов использовал очередь с приоритетом, построенной на векторе и упорядочиванием по возрастанию.

Сам алгоритм: на каждой итерации основного цикла выбирается вершина, которой на текущий момент соответствует

минимальная оценка кратчайшего пути. Производится релаксация всех исходящих из неё рёбер. Сложность  $O(m \cdot \log(n))$ .

#### **Алгоритм Форда-Беллмана:**

Для алгоритма удобнее хранить граф в векторе рёбер, как я и поступил. Результат всё так же в `vector<long long>`. Для поиска отрицательных циклов нужен флаг, говорящий об изменении какого-то ребра.

Сам алгоритм: на каждой фазе просматриваются все рёбра графа, и алгоритм пытается произвести релаксацию. Фактически это значит, что мы пытаемся улучшить ответ для вершины  $b$ , пользуясь ребром  $(a,b)$  и текущим ответом для вершины  $a$ . Сложность  $O(n \cdot m)$ .

#### **Алгоритм Джонсона:**

Необходимы алгоритмы Форда-Беллмана и Дейкстры, а также метод перевзвешивания.

Сам алгоритм: считав граф дополняем его фиктивной вершиной, имеющей ребра во все остальные вершины. Ищем с помощью алгоритма Форда-Беллмана минимальные расстояния до всех вершин. Используем эти расстояния для перевзвешивания, которое уберёт из графа рёбра отрицательной длины. Запускаем алгоритм Дейкстры для полученного графа и опять перевзвешиваем, чтобы вывести ответ. Сложность  $O(n \cdot m + n \cdot m \cdot \log(n))$ .

#### **Тест программы:**

Количество вершин	количество ребер	время выполнения
500	500	3134
1000	500	8439
500	1000	6241
1000	1000	11850

### **Дневник отладки:**

Делал я алгоритмы начиная с Дейкстры. Сначала столкнулся с ограничением по памяти и переписал код с матрицы смежности на вектор смежности. Потом столкнулся с ограничением по времени. Пришлось переписать алгоритм поиска следующей для посещения вершины.

В алгоритмах Форда-Беллмана и Джонсона не пришлось отлаживать.

### **Выводы**

Выполнив восьмую лабораторную работу по курсу "Дискретный анализ я разобрался в теме графов, решил задачу на эту тему, а также потренировался в использовании разнообразных стандартных контейнеров.