

3 задача: Количество различных путей

1 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

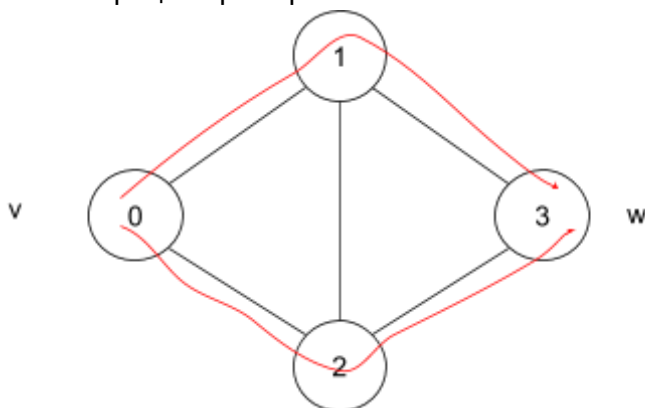
Дан невзвешенный неориентированный граф. В графе может быть несколько кратчайших путей между какими-то вершинами. Найдите количество различных кратчайших путей между заданными вершинами. Требуемая сложность $O(V+E)$.

Ввод: v : кол-во вершин (макс. 50000), n : кол-во ребер (макс. 200000), n пар реберных вершин, пара вершин v, w для запроса.

Вывод: количество кратчайших путей от v к w

in	out
4	2
5	
0 1	
0 2	
1 2	
1 3	
2 3	
0 3	

Иллюстрация примера:



2. Описание алгоритма

Запускаем BFS из начальной вершины. Для каждой посещённой вершины динамически вычисляем количество способов попадания в эту вершину. Как только мы доходим до конечной вершины, прерываем алгоритм

3. Доказательство корректности

Докажем, что правильное количество путей в некую вершину будет записано по завершению итерации (“слоя” расстояний), в которой эта вершина была обнаружена и положена в очередь. При обработке вершин с большим расстоянием до source это число изменяться не будет

Пусть от начальной вершины v до некоторой вершины u расстояние d . Выберем момент времени когда в очереди находятся все вершины, от которых расстояние до начальной вершины $d-1$ (далее такие вершины будем обозначать V_{d-1}) и только они (такой момент времени всегда найдётся в связи с принципом работы BFS). Из

некоторых вершин V_{-1} есть рёбра в u . После обработки всех вершин из V_{-1} в $paths[u]$ будет записано количество путей длины d через вершины V_{-1} . После этого начнётся обработка вершин V_0 (с расстоянием d до v). Но если из этих вершин есть рёбра в u , то они уже будут частью пути длины $d+1$, то есть это будут уже не кратчайшие пути. Далее это расстояние будет только увеличиваться, значит после обработки всех V_{-1} количество кратчайших путей уже было найдено

Из доказанного индуктивно следует, что после обнаружения конечной вершины можно перестать помещать следующие вершины в очередь. Ответ уже будет найден после обработки вершин, лежащих в очереди до обнаружения конечной вершины

4. Время работы и дополнительная память

$$T = O(V + E)$$

$$M = O(V)$$

5. Доказательство времени работы и дополнительной памяти

а) Время работы:

Алгоритм состоит из одного запуска BFS, время работы которого составляет

$$T = O(V + E)$$

б) Дополнительная память:

Размер контейнеров для: очереди, буфера для следующих вершин, расстояний, количества путей - не превышает V , значит $M = O(V)$