# Домашнее Задание по алгоритмам №10

Павливский Сергей Алексеевич , 873

14.04.2019

## Задача 1.

В графе может быть несколько кратчайших путей между какими-то вершинами. Постройте линейный по времени алгоритм, находящий количество вершин, которые лежат хотя бы на одном кратчайшем пути из s в t в неориентированном графе с единичными весами на рёбрах.

### Решение

Алгоритмом DFS за O(|V| + |E|) найдем картчайший путь из вершины $s$ в вершину $t$ . Так как каждое ребро имеет вес 1 , то длина пути будет равна количеству вершин - 1 ( если учитывать вершины $s$ и $t$) . Тогда мы за линейное время = O(|V| + |E|) нашли количество вершин на кратчайшем пути , что и требовалось .

## Задача 2.

Рассмотрим следующую модификацию алгоритма Дейкстры. При инициализации, в очереди с приоритетами находится лишь вершина s. Вершина v добавляется в очередь с приоритетами, если в результате релаксации Relax(u, v) расстояние до вершины v изменилось, и при этом v не была в этот момент в очереди. Остальные шаги алгоритма остались без изменений.

1. Докажите корректность модифицированного алгоритма.

Заметим , что расстояние до ранее удаленных вершин не изменяется , а значит никакая вершина вновь не появится в куче . А так же заметим , что $+\infty$ не влияют на минимальный элемент в куче , если там есть хоть один элемент с конечным значением , но если вершины с конечными значениями кончились , то все остальные вершины недостижимы , а значит до них расстояние на самом деле $+\infty$ .

2. Докажите, что модифицированный алгоритм работает корректно даже в случае наличия рёбер отрицательного веса, но при отсутсвии цикла отрицательного веса. Оцените время работы алгоритма на графах такого вида и сравните его со временем работы алгоритма Беллмана-Форда.

3. Модифицируйте алгоритм так, чтобы он выдавал ошибку на графах с циклами отрицательного веса.

# Задача 3.

Профессор О. П. Рометчивый предлагает следующий способ нахождения кратчайшего пути из s в t в данном ориентированном графе, содержащем рёбра отрицательного веса. Прибавим достаточно большую константу к весам всех рёбер и сделаем все веса положительными, после чего воспользуемся алгоритмом Дейкстры. Корректен ли такой подход? Если да, то докажите это, если нет — укажите контрпример.

## Решение

Данный способ некорректен . Пусть исходно в графе было 2 пути из вершины $s$ в вершину $t$ : 1 , -1 и 1 . Тогда путь 1 будет кратчайшим , т.к. 1 - 1 = 0 , 0 < 1. Но если прибавить ко всем весам константу , что все веса станут >= 0 , то путь 1 станет не меньше чем путь 2 , так как он будет содержать те же неотрицательные компоненты , что и путь 2 , а так же некоторую неотрицательную величину , в которую превратится -1 . Но тогда путь 1 уже будет не кратчайшим , а значит алгоритм будет работать некорректно .

# Задача 4.

Предложите $O(|V|+|E|)$ алгоритм поиска кратчайших расстояний от данной вершины s до всех остальных в графе , в котором все веса ребер равны 0 или 1. Докажите его корректность и оцените асимптотику.

## Решение

Для каждой вершины будем хранить номер уровня , на котором она находится , ведя две очереди : вершины , рассматриваемые на текущем шаге и вершины , которые будут рассматриваться на следующем шаге.

При добавлении очередной вершины во вторую очередь будем добавлять в нее же все еще не рассмотренные вершины , расстояние до которых от данной 0 . Реализуем это с помощью поиска в ширину , где мы ходим только по нулевым ребрам .

Изначально для каждой вершины номер уровня $+\infty$ .

Ответом для каждой вершины будет номер уровня , на котором она находится .

Ассимптотика :

Каждая вершина рассматривается не больше 1 раза , а каждое ребро — не больше двух раз (со стороны одного и другого конца) . Итого : $O(|V| + |E|)$ , что и требовалось .

Корректность :

Доказательство аналогично доказательству корректности самого поиска в ширину :

Рассмотрим момент времени , когда i-ый шаг закончился . Мы уже сформировали $(i + 1)$ слой . Вершины , находящиеся во второй очереди , — все вершины , которые будут на $(i + 1)$ - ом слое . Заметим , что все ребра , ведущие из них в еще не рассмотренные вершины имеют вес 1 по алгоритму . А так же заметим , что если есть путь от одной вершины до другой длинной 0 , то эти вершины находятся на одном расстояние от изначальной .

То есть и вправду на i - ом слое будут все вершины , расстояние до которых i .

# Задача 5.

В орграфе есть ребра отрицательного веса, но нет циклов с отрицательным весом. Предложите алгоритм, который находит для данной вершины вершину, от которой она удалена на максимальное расстояние. Докажите его корректность и оцените асимптотику.

## Решение

Так как нет отрицательных циклов , то мы можем использовать алгоритм Форда-Беллмана . Модифицируем алгоритм Форда-Беллмана так : d[k + 1][v] = max(d[k + 1][v], d[k][u] + $\omega$(u,v)) ( строчка из реализации алгоритма на neerc.ifmo.ru ), где $\omega$(u,v) - вес ребра uv . Тогда данная модификация будет считать максимальное расстояние от данной вершины то остальных . После окончания работы алгоритма за |V| пройдемся по всем вершинам и найдем максимальное расстояние . Эта вершина и будет ответом .

Ассимптотика : Форд-Беллман + $|V| = O(|V| \cdot |E|) + |V| = O(|V| \cdot |E|)$ .

Корректность : модифицированный алгоритм Форда-Беллмана найдет максимальные расстояния от данной вершины до остальных (доказательство аналогично доказательству корректности работы обычного алгоритма Форда-Беллмана) , впоследствие среди них будет выбрана максимальная . Максимум среди максимальных расстояний очевидным образом будет максимальным расстоянием .

# Задача 6.

Независимое множество в неориентированном графе — это множество вершин попарно не соединенных ребрами. Предложите $O(|V| + |E|)$ алгоритм поиска максимального по размеру независимого множества в дереве.

## Решение

Рассмотрим представление графа в виде дерева . Из структуры дерева очевидно , что если две вершины лежат на разных уровнях дерева , находящиеся на расстоянии не меньше одного уровня , то они принадлежат независимому множеству . Также из структуры дерева очевидно , что вершины на одном уровне принадлежат независимому множеству . Тогда пройдемся по графу алгоритмом DFS из вершины дерева. После того как мы обрабатываем вершины с одного уровня в дереве , у нас последующим цельным блоком в стеке лежат вершины следующего уровня в дереве ( так как у нас последовательно лежат потомки элементов предыдущего блока из стека ) . Для каждой вершины из последовательно заносящегося блока вершин ( исходный блок в стеке - корень дерева , далее каждый блок - потомки предыдущего блока ) будем хранить четность блока , которому она принадлежит . В переменной sum1 будем хранить количество вершин из блоков с четными номерами , в переменной sum2 количество вершин из блоков с нечетными номерами . После окончания алгоритма ответом будет max(sum1, sum2) .

Ассимптотика : на каждом шаге DFS мы делаем константное количество операций , т.е. сложность алгоритма = сложности DFS = $O(|E| + |V|)$ что и требовалось .

Корректность :

Максимальное количество достигается , если брать уровни через 1 ( пропуская уровень мы ничего не выигрываем , а просто рассматриваем более проигрышный случай выбора уровней другой четности , так как мы попросту недобрали часть уровней , т. к. из структуры дерева количество на каждом последующем уровне $\geqslant$ количества вершин на предыдущем ) .

Докажем по индукции .

База : 2 уровня , корень и его потомки - очевидно выполняется .

Индукционный переход : пусть выполняется для n уровней . Тогда при добавлении n + 1 он может оказаться в одном независимом множестве с вершинами с уровнем той же четности . Если ри его добавлении количество вершин с четностью его уровня оказалось больше количества вершин с другой четностью уровня , то ответ изменит свое значение , иначе окажется прежним , т. е. при добавлении нового уровня алгоритм остается корректен , то есть он корректен всегда .