Домашнее задание по АМВ

771 группа, Христолюбов Максим

13 апреля 2019 г.

1 Задача 1

Будем использовать несколько последовательных поисков в ширину. Так как он занимает $O(n+m)$ времени, то весь алгоритм $O(n+m)$.

Найдем поиском в ширину расстояние от a до v_1, а потом до v_2, а так же от v_1 и v_2 до b. После этого поиском в ширину найдем расстояние от a до b. Если $S(a,b) = S(a,v_1)+S(v_1,b)+1$ или $S(a,b) = S(a,v_2)+S(v_2,b)+1$, то ребро v_1v_2 принадлежит одному из кратчайших путей от a до b, иначе — не принадлежит.

2 Задача 2

Можно либо предварительно стянуть все ребра, которые имеют вес 0, то есть вместо 2 вершин, соединенных ребром веса 0, поставить одну вершину, которая будет соединяться со всеми вершинами, с которыми были соединены изначальные 2 вершины. Тогда в графе не будет ребер 0 длины и можно будет запустить BFS.

Или можно каждый раз перед тем как находить вершины, находящиеся на расстоянии $k+1$, рассматривая соседей вершин, которые находятся на расстоянии k, запускать поиск в глубину из каждой вершины, которая на расстоянии k, по ребрам 0 длины и добавлять найденные вершины ко множеству вершин, находящихся на расстоянии k.

3 Задача 3

4 Задача 4

4.1 a)

Алгоритм Флойда-Уоршелла итак корректно работает в графах с отрицательными ребрами без отрицательных циклов. Если же в графе есть отрицательный цикл, то для какого-то i $d_{ii}^{(}n) < 0$, но в этих графах задача о поиске кратчайшего пути не имеет смысла.

4.2 б)

$$D^{(0)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}, D^{(2)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}, D^{(4)} = \begin{pmatrix} 0 & 1 & 3 & -1 \\ \infty & 0 & \infty & -2 \\ \infty & -2 & 0 & -4 \\ \infty & \infty & \infty & 0 \end{pmatrix}$$

5 Задача 5

5.1 b)

Метки перестанут обновляться после k-ой фазы алгоритма, где k — кол-во ребер в кратчайшем пути с наибольшим кол-вом ребер в нем, так как до этой фазы этот кратчайший путь еще не был найден, а после этой фазы все кратчайшие пути длины меньше-равных k уже найдены, то есть найдены все кратчайшие пути и метки обновляться не будут. Перебором и внимательным вглядыванием кол-во ребер в кратчайшем пути до D — 0, C — 1, E — 2, B — 2, H — 2, G — 3, F — 4, A — 3, значит, после 4 фазы метки обновляться не будут.

5.2 c)

Получается, что наибольшая реберная длина кратчайшего пути до некоторой точки в графе равна $|V| - 1$. Так как веса всех ребер равны 1, то реберная длина равняется длине кратчайшего пути. Так как всего вершин $|V|$, то все они должны входить в этот самый длинный кратчайший путь между 2 точками, причем в этом пути, конечно, нет самопересечений, следовательно,граф представляет из себя цепочку из $|V|$ вершин с $|V| - 1$ ребрами между ними.

6 Задача 6

Их количество будет равно количеству вершин. Так как из определения частичного порядка следует, что aRb, bRa следует, что $a = b$, а если a и b достижимы друг из друга, то это значит, что существует такая последовательность вершин c_1,$c_2 \ldots c_k$, что есть двусторонние ребра $(a$,$c_1)$,$(c_1$,$c_2) \ldots (c_{k-1}$,$c_k)$,$(c_k$,$b)$, значит, так как на графе определено отношение порядка из условия, то aRb и bRa, следовательно, $a = b$. Отсюда следует, что все вершины из одной компоненты сильной связности совпадают, то есть в каждой компоненты сильной связности только одна вершина.

7 Задача 7

7.1 a)

Пусть при поиске в глубину вход в вершину u произошел раньше, тогда прежде чем выйти из u поиск в глубину дойдет до вершины v и выйдет из нее, и время выхода из u будет больше чем из v.

Пусть вход в вершину v произошел раньше, тогда так как u не достижима из v, то поиск в глубину выйдет из вершины v, так и не войдя в вершину u. Позже поиск в глубину дойдет до вершины u и потом выйдет из нее, и время выхода из u будет больше чем из v.

7.2 b)

Если бы ребро графа шло бы от вершины v с меньшим номером к u с большим номером, тогда это значило бы, что существует такой поиск в глубину, что время выхода из u больше, чем время выхода из v, но u достижимо из v, и v не достижимо из u, так как граф ациклический, значит, как было доказано ранее время выхода из v больше, чем время выхода из u для любого поиска в глубину — противоречие. Значит, такого быть не может и ребра идут от вершин с большим номером к меньшим.

8 Задача 8

8.1 a)

При инвертации графа достижимость u из v заменяется на достижимость v из u. Если $u \sim v$, то они оба достижимы из друг друга, и при инвертации они останутся быть достижимыми друг из друга, то есть $u' \sim v'$. Значит, все вершины, находящиеся в компонентах сильной связности останутся в них же, компоненты сильной связности не изменяться.

8.2 b)

Для инвертации графа потребуется $O(m)$, для поиска в глубину в инвертированном графе потребуется $O(n+m)$, для поиска в глубину в изначальном графе еще $O(n+m)$. Всего $O(n+m)$.

Для алгоритма требуется инвертировать граф, но можно не создавать новый граф, а просто инверсно интерпретировать элементы матрицы смежности. При поиске в глубину требуется хранить вершины в стеке, в худшем случае в нем будут все n вершин. Так же требуется запоминать время выхода для каждой вершины, еще $O(n)$ памяти. Отсортировать вершины можно используя константную дополнительную память в дополнение к памяти занимаемой самим сортируемым множеством. Значит всего уйдет памяти $O(n)$.

8.3 c)

Да, так как в конденсации графа нет циклов (является деревом, в общем случае лесом), то всегда можно отсортировать вершины, чтобы ребра шли только от вершин с большим номером к меньшим.