Динамическое транзитивное замыкание

Екатерина Рыбина, 371 гр.

1

Придумайть алгоритм для декрементального транзитивного замыкания, работающий за $O(n^2(m+n))$ суммарно на все обновления.

Решение:

Algorithm 1

function ADD(i, j)
    $A(i).del(j)$
    $A^T(j).del(i)$
    $visited = DFS(A, i)$
    for $v : v \notin visited$ & $M(j, n) = 1$ **do**
        $visited^T = DFS(A^T, v)$
        for $u : u \notin visited^T$ & $M(u, v) = 1$ **do**
            $M(u, v) = 0$
        end for
    end for
end function

Пояснение:
После удаления ребра (i, j) необходимо обработать такие вершины v, которые достижимы из j, но стали недостижимы из i. Для этого помимо исходной матрицы достижимости(M) до начала работы алгоритма, мы хранить список смежности для исходного графа (A).
Прим.: Для списка смежности используется обозначение для матрицы смежности. Это сделано чисто ради удобства.
Когда удаляем ребро (i, j), обновляем A и запускаем поиск в глубину из вершины i - $DFS(A, i)$.
Однако есть вершины, из которых достижимы i и v. Нам нужны только те, из которых v не достижима после удаления ребра (i, j). Это можно

проверить, запустив DFS из v на инвертированном графе. Если есть путь из v в u, то и в исходном графе есть пусть из u в v. Таким образом нам понадобиться список смежности для обратного графа - A^T.

Сложность:

Так как мы удаляем m рёбер, то и алгоритм запуститься m раз.
После проверки в строке (5) в наихудшем случае строки (6)-(8) запустятся n раз. Из-за DFS общая сложность алгоритма после удаления одного ребра $O(n(n+m))$.
Таким образом сложность всех обновлений после удаления m рёбер и учитывая что m не превосходит n^2 - $O(n^3(n+m))$.
Однако мы только удаляем рёбра и потому (5)-(8) выполняются не более n^2 раз за все обновления. В итоге получаем $O(n^2(n+m))$.