Динамическое транзитивное замыкание

Иванов Кирилл
371 группа

Задача 1. На лекции мы научились поддерживать инкрементальное транзитивное замыкание ориентированных графов. Придумайте алгоритм для декрементального транзитивного замыкания, работающий за $O(n^2(m+n))$ суммарно на все апдейты.

Решение. При удалении нас будут интересовать лишь те вершины, из которых i достижима, поскольку лишь для них может потребоваться обновление информации в матрице. Из каждой такой вершины u запустим поиск в глубину, при помощи которого определим вершины достижимые из u – их изменять не требуется. Пусть вершина v недостижима из u (после удаления ребра (i, j)), при этом $M[j, v] = 1$, тогда нам необходимо занулить путь между u и v. Для работы поиска в глубину будем также поддерживать исходный граф в виде списков смежности.

Algorithm 2 Декрементальное обновление матрицы достижимости

```
1: function REMOVE(i, j)
2:     g[i].remove(j)
3:     M[i, j] = 0
4:     for u ≠ j : M[u, i] = 1 do
5:         dfs(u)                                  ▷ Помечает в used достижимые вершины
6:         if ¬used[j] then
7:             for v : ¬used[v] ∧ M[j, v] = 1 do
8:                 M[u, v] ← 0
```

Анализ сложности. Поскольку алгоритм декрементальный, если $M[i, j]$ стало 0, то оно никогда не станет 1. Четвертая строка будет стоить $O(n)$ для одного удаления, строки 5–8 – $O(T_{dfs}(n, m) + n) = O(n + m + n) = O(n + m)$. Заметим, что строки 5–8 будут вызваны не более $O(n^2)$ раз, то есть итоговое время работы – это $O(n^2(n + m))$.