Студент: Милосердова Любовь Михайловна
Группа: 371 (SE)
Дата: 5 мая 2022 г.

Динамическое транзитивное замыкание

Задача 1. *Придумайте алгоритм для декрементального транзитивного замыкания, работающий за $O(n^2(m+n))$ суммарно на все апдейты.*

Доказательство. Будем хранить список смежности графа и удалять ребра из него. Также будем хранить граф, в котором напревления ребер заменены на противоположные. Затем запустим обход в глубину из вершины i, тем самым найдем вершины, достижимые из i. Если вершина u достижима из i в "перевернутом"графе, а вершина v не достижима из i в исходном графе, значит между вершинами u и v нет пути, то есть необходимо занулить соответствующую ячейку в матрице достижимости.

Псевдокод алгоритма:

Algorithm 1 Декрементальное обновление матрицы достижимости

```
function DELETE(i, j)
    if G[i, j] = 1 then
        G[i].remove(j)                    ▷ G — список смежности графа
        G_reversed[j].remove(i)           ▷ G_reversed — список смежности графа, в котором направление ребер изменено на проотивоположное
        dfs(G_reversed, i)                ▷ сохраняет в массив used_reversed вершины, из которых достижима i
        dfs(G, i)                         ▷ сохраняет в массив used достижимые из i вершины
        for u : used_reversed[u] = 1 do
            for v : used[v] = 0 do
                M[u, v] ← 0
```

Сложность алгоритма обхода графа в глубину — $O(m+n)$. Тогда суммарная сложность нашего алгоритма при удалении всех ребер — $O(m(m+n+n^2))$, так как m раз вызывается функция *delete*. Так как количество ребер не превышает количества вершин в квадрате, получаем $O(m(m+n)+n^2m) \leq O(n^2(m+n)+n^2m) = O(n^2(m+n))$ □