

# Динамическая связность

Мирошников Владислав

371 группа

17.05.2022

**1 (a).** Придумайте рекурсивную процедуру  $fall(v)$ , которая для вершины  $v$ , такой, что  $N_1(v) = \emptyset$ , "роняет"  $v$  на правильный уровень BFS-дерева, корректно обновляет уровни соседей  $v$  и "роняет" те вершины, чей уровень изменился при падении  $v$ .

*Решение*

Считаем, что перед вызовом рекурсивной процедуры  $fall(v)$  выполняется проверка на непустоту  $N_2(v)$ . В противном случае будет образовываться новая компонента связности.

---

**Algorithm 1**

---

```
1: function FALL( $v$ )
2:    $l(v) \leftarrow l(v) + 1$ 
3:   for  $u \in N_2(v)$  do
4:      $N_2(u) \leftarrow N_2(u) \setminus \{v\}$ 
5:      $N_3(u) \leftarrow N_3(u) \cup \{v\}$ 
6:   for  $u \in N_3(v)$  do
7:      $N_1(u) \leftarrow N_1(u) \setminus \{v\}$ 
8:      $N_2(u) \leftarrow N_2(u) \cup \{v\}$ 
9:    $N_1(v) \leftarrow N_2(v)$ 
10:   $N_2(v) \leftarrow N_3(v)$ 
11:   $N_3(v)_{old} \leftarrow N_3(v)$ 
12:   $N_3(v) \leftarrow \emptyset$  ▷ При вызове  $fall$  от детей происходит заполнение
13:  for  $u \in \{w \mid w \in N_3(v)_{old} \text{ and } N_1(u) = \emptyset\}$  do
14:     $fall(u)$ 
```

---

**1 (b).** Докажите, что если в графе  $n$  вершин и  $m$  рёбер изначально, на все обновления суммарно при удалении  $m$  рёбер уйдёт время  $O(mn)$ .

*Доказательство.* Положим степень вершины  $v := deg(v)$ . Обработка одной вершины внутри рекурсивной процедуры  $fall(v)$  занимает  $O(deg(v))$  времени. Далее оценим максимальное количество вызовов процедуры  $fall$ , инициированных вершиной  $v$ . В случае, если BFS-дерево почти вырождается в список, то есть его высота сравнима с  $n$ , то максимальное количество вызовов процедуры сравнимо с данной величиной  $n$ . Таким образом, работа процедуры  $fall(v)$  для вершины  $v$  занимает  $O(n deg(v))$  времени, а удаление ребра занимает  $O(1)$  времени. В итоге получаем следующее выражение:

$$O(m + \sum_{v \in V} n deg(v)) = O(m + n \sum_{v \in V} deg(v)) = O(m + nm) = O(mn) \quad \square$$

**1 (c).** Пусть вместо всего BFS-дерева нам разрешено хранить только BFS-дерево с  $d$  уровнями, т.е. структура будет поддерживать только расстояния до вершин  $v$ , такие, что  $d(s, v) \leq d$ . Докажите, что суммарное время на все апдейты в этом случае равно  $O(md)$ .

*Доказательство.* Положим степень вершины  $v := deg(v)$ . Далее оценим максимальное количество вызовов процедуры  $fall$ , инициированных вершиной  $v$ . В случае, если BFS-дерево почти вырождается

в список, то есть его высота сравнима с  $d$  ( $d$  — количество уровней BFS-дерева), то максимальное количество вызовов процедуры сравнимо с данной величиной  $d$ . Таким образом, работа процедуры  $fall(n)$  для вершины  $v$  занимает  $O(d \deg(v))$  времени, а удаление ребра занимает  $O(1)$  времени. В итоге получаем следующее выражение:

$$O(m + \sum_{v \in V} d \deg(v)) = O(m + d \sum_{v \in V} \deg(v)) = O(m + dm) = O(md) \quad \square$$