Динамическая связность

Кузнецов Илья Александрович
371 группа

13.05.2022

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

Решение. Положим, что перед вызовом рекурсивной процедуры $fall(v)$: $N_2(v) \neq \varnothing$. В противном случае будет образована новая компонента связности.

Algorithm 1

function FALL(v)
  $l(v) \leftarrow l(v) + 1$
  for $u \in N_2(v)$ **do**
    $N_2(u) \leftarrow N_2(u) \setminus \{v\}$
    $N_3(u) \leftarrow N_3(u) \cup \{v\}$
  for $u \in N_3(v)$ **do**
    $N_1(u) \leftarrow N_1(u) \setminus \{v\}$
    $N_2(u) \leftarrow N_2(u) \cup \{v\}$
  $N_1(v) \leftarrow N_2(v)$
  $N_2(v) \leftarrow N_3(v)$
  $N_3(v)_{old} \leftarrow N_3(v)$
  $N_3(v) \leftarrow \varnothing$ ▷ Заполняется при вызове $fall$ от вершин-детей
  for $u \in \{w \mid w \in N_3(v)_{old}$ and $N_1(u) = \varnothing\}$ **do**
    $fall(u)$

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

Доказательство. Пусть $deg(n)$ — степень вершины v. Обработка одной вершины внутри описанной рекурсивной процедуры при этом будет занимать $O(deg(n))$ времени.

В случае, когда BFS-дерево вырождается в список, его высоту можно сравнить с n. Тогда наибольшее число вызовов рекурсивной процедуры $fall$, начатых из вершины v, тоже можно сравнить с n. Отсюда, работа процедуры $fall$ для вершины v занимает $O(n\ deg(v))$ времени. Удаление ребра занимает $O(1)$ времени.

Таким образом, имеем: $O(m + \sum_{v \in V} n\ deg(v)) = O(m + n \sum_{v \in V} deg(v)) = O(m + nm) = O(mn)$. □

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 обновления в этом случае равно $O(md)$.

Доказательство. Пусть $deg(n)$ — степень вершины v. Обработка одной вершины внутри описанной рекурсивной процедуры при этом будет занимать $O(deg(n))$ времени.

В случае, когда BFS-дерево вырождается в список, его высоту можно сравнить с d. Тогда наибольшее число вызовов рекурсивной процедуры $fall$, начатых из вершины v, тоже можно сравнить с d. Отсюда, работа процедуры $fall$ для вершины v занимает $O(d\ deg(v))$ времени. Удаление ребра занимает $O(1)$ времени.

Таким образом, имеем: $O(m + \sum_{v \in V} d\ deg(v)) = O(m + d \sum_{v \in V} deg(v)) = O(m + dm) = O(md)$. $\square$