Динамическая связность

Екатерина Рыбина, 371 гр.

1 1(a)

Придумать рекурсивную процедуру *fall(v)*, которая 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:

Algorithm 1

procedure FALL(v)
  $l(v) \leftarrow l(v) + 1$
  for $u \in N_2(v)$ **do**
    $N_2(u) \leftarrow N_2(u) \setminus v$
    $N_3(u) \leftarrow N_3(u) \bigcup v$
  end for
  for $u \in N_3(v)$ **do**
    $N_1(u) \leftarrow N_1(u) \setminus v$
    $N_2(u) \leftarrow N_2(u) \bigcup v$
  end for
  $N_1(v) \leftarrow N_2(v)$
  $N_2(u) \leftarrow N_3(u)$
  for $u \in N_2(v) \&\& N_1(u) = \emptyset$ **do**
    FALL(u)
  end for
end procedure

Пояснение:
Предполагаем, что перед запуском *fall(v)* мы проверили граф на появление новой компоненты связности после удаления ребра. Если вершины не принадлежат компоненте, в которой находится стартовая вершина, то до них расстояние будем считать бесконечным.

(3)-(6): Все соседи вершины v, которые были с ней на одном уровне, становятся родителями.
(7)-(10): Вершины, для которых v была родителем, теперь становятся на один уровень с ней.
(11)-(12): Делаем соответствующие преобразования для вершины v.
(13)-(15): Запускаем рекурсивно для детей v, которые всё ещё на одном с ней уровне, но нет родителя, чтобы тоже их опустить.

2 1(b)

Доказать, что если в графе n вершин и m рёбер изначально, на все обновления суммарно при удалении m рёбер уйдёт время *O(mn)*.
Док-во:
При работе алгоритма для одной вершины v нам нужно будет обработать всех её соседей, количество которых *deg(v)* - степень вершины v. То есть время работы алгоритма тогда будет *O(deg(v))*. В худшем случае $deg(v) = n$, что может соответствовать ситуация, когда после удаления ребра граф вытянется в одну ветку и обновить надо все n вершин. А так как мы удаляем m ребер, то и алгоритм запускаем m раз. В итоге это даёт нам время *O(mn)*.
ч.т.д

3 1(c)

Пусть вместо BFS-дерева нам разрешено хранить только BFS-дерево с d уровнями, то есть структура будет поддерживать только расстояния до вершин v такие что $d(s, v) \leq d$. Доказать, что суммарное время на все обновления в этом случае равно *O(mn)*.
Док-во:
Аналогично предыдущей задаче, но теперь у нас в худшем случае обработка одной вершины будет занимать *O(d)*. Тогда при удалении m рёбер, суммарное время для всех обновлений равна *O(dm)*.
ч.т.д