Домашняя работа №3

Тетерин Дмитрий, Б05-932

16.04.2020

Задача 1. Предложите метод слияния двух AVL-деревьев за O(log(|T1| + |T2|)) (все ключи левого дерева меньше всех ключей правого).

Решение. Приведём алгоритм и будем на каждом его шаге доказывать, что мы не выходим за предложенную ассимптотику.
1. Определим высоту сливаемых деревьев. В моей реализации AVL-деревьев (см. контест) это $O(1)$.
2. Удалим самый правый элемент α из левого дерева (либо самый левый из правого, если левое дерево оказалось выше), предварительно запомнив его. $O(log|T1|)$
3. В правом (или, соответственно, левом, если левое было выше) дереве будем идти влево (вправо), пока не найдём элемент β, в котором поддерево не более чем на 1 выше левого (правого). $O(log|T2|)$
4. Заменяем $\beta_{new} \to key = \alpha \to key; \beta_{new} \to left = left\_tree; \beta_{new} \to right = \beta$. $O(1)$
5. Увеличиваем баланс родителя. Сама β уже сбалансирована. $O(1)$
6. Восстанавливаем баланс, как сделали бы после обычной вставки β. $O(log(|T1|+|T2|)$.
Таким образом, мы слили два AVL-дерева с ассимптотикой $O(log(|T1|+|T2|))$ и задача решена. □

Задача 4. Пусть дано декартово дерево. Не ухудшая асимптотику стандартных операций, находясь в вершине, предложите метод поиска следующей в естественном порядке. Более формально, если алгоритм находится в вершине v с ключом x, как попасть в вершину u с минимальным ключом z > x? Асимптотика ответа на запрос:
а) O(log n) в среднем;
б) O(1).

Решение. а) Когда мы находимся в вершине с ключом x, найдём самый левый элемент её правого поддерева. Т.к. ~~курево~~ Декартово дерево остаётся обычным деревом поиска, этот алгоритм даст нам ответ за $O(logn)$.
б) Для асимптотики $O(1)$ будем глобально в дереве поддерживать указатели на начало и конец двусвязного списка, а в каждой вершине дерева – указатели на следующий и предыдущий элементы этого списка, отсортированного по возрастанию ключей. Тогда, находясь в вершине с ключом x, мы находим ответ за $O(1)$.
Поддержание такого списка обеспечивается простым фиксом функции вставки в дерево: если вставляемый элемент больше родителя, то вставим его между родителем и его последующим элементом, иначе – между родителем и его предыдущим элементом. Стоимость этого фикса - $O(1)$. Аналогично с удалением. Асимптотика стандартных операций с деревом не ухудшилась.

$\square$

Задача 5. Дано взвешенное дерево, то есть на каждом ребре написано некоторое целое (возможно, отрицательное) число. Найдите в нём простой путь с наибольшей суммой. Асимптотика: O(n), где n – число вершин в дереве.

Решение. □