

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ ДЛЯ СТУДЕНТОВ

Деревья.

В задачах этой серии, n – количество элементов в дереве, h – высота дерева (максимальное расстояние от корня дерева до листа).

1. Реализуйте функцию, которая бы выводила все элементы двоичного дерева поиска. Время $O(n)$.
- 2.* Реализуйте функцию, которая бы выводила все элементы двоичного дерева поиска. Время $O(n)$, память $O(1)$. Считайте, что в каждой вершине хранится ссылка на родителя. Помните, что на каждый рекурсивный вызов тратится память.
3. Научитесь находить k -й в порядке возрастания ключ в двоичном дереве поиска за $O(h)$. Разрешается хранить в каждой вершине $O(1)$ дополнительной памяти.
4. Научитесь по данному ключу x находить количество элементов в дереве, не превосходящих x за $O(h)$. Разрешается хранить в каждой вершине $O(1)$ дополнительной памяти.
5. По заданной вершине в дереве найдите следующие k вершин в порядке сортировки за $O(h + k)$.
6. В AVL-дереве мы поддерживаем инвариант, что для каждой вершины высота её детей отличается не более, чем на 1. Тогда верно, что $h = O(\log n)$. Правда ли, что $h = O(\log n)$, если:
 - (a) для каждой вершины высота ее детей отличается не более, чем на 5;
 - (b) для каждой вершины высота ее детей отличается не более, чем в 2 раза.
7. Научитесь при помощи двоичного дерева поиска отвечать на следующие запросы за $O(h)$:
 - (a) Добавить число x в множество;
 - (b) Вычислить сумму всех x из множества, таких что $l \leq x \leq r$.