Тураев Тимур, 504 (SE)
18.02.2014

1.2. Доказать, что в AVL-дереве можно вывести k подряд идущих элеметов за O(k+logN). Дополнение про размер «лишних» узлов.

Нужно заметить, что последовательный вызор SUCC (начиная с некоторого узла X) из посещенных узлов формирует дерево. В этом дереве, как я уже писал, есть ровно k узлов, которые входят в искомое множество «k подряд идущих», а также другие узлы, количество которых можно ограничить сверху значением 2(h+1) = O(h) = O(logN): почему? Посмотрим как получается это дерево: оно очень похоже на обычный in-order обход AVL-дерева, только «начинается» оно с узла X (не исключено, что в дереве будут узлы со значением меньшим X) и «заканчивается» на k-ом узле после X (но опять же не исключено, что в дереве не будет узлов со значением больше X); как раз такие узлы назовем лишними. Если еще раз посмотреть **где** будут расположены такие узлы, то можно заметить, что они будут лежать либо на крайней левой, либо на крайней правой ветви «дерева поиска k элементов», значит их число ограничено двумя высотами дерева, то есть значением 2(h+1) = O(h) = O(logN).

1.3. Сохранить высоту вершины в AVL-дереве за O(1) бит.

…
Высота вершины будет вычисляться так:

```
h(node) = (node.characteristic == 01 {левое поддерево больше}) ? h(node.left) + 1 : h(node.right) + 1
```

Если внимательно посмотреть, то фактически мне нужно отличать ситуацию «левое поддерево больше» от «левое поддерево не больше», поэтому да, достаточно **одного** бита.