Feb 4, 2016

Recall Heap / Priority Queue

You can build a heap in $\Theta(n)$ if all the elements are given in a array before hand.

BST — Binary Search Tree $[a, b]$..

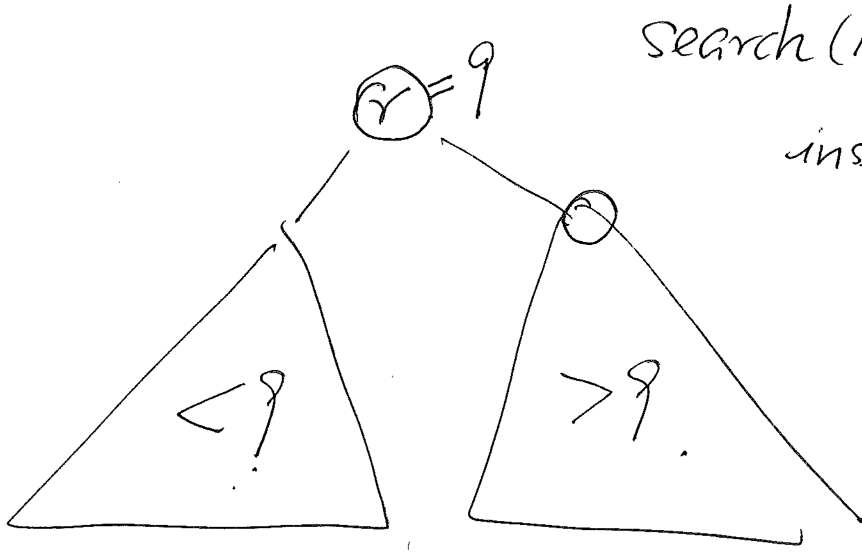
Heap $[a, +\infty)$ $(-\infty, a]$.

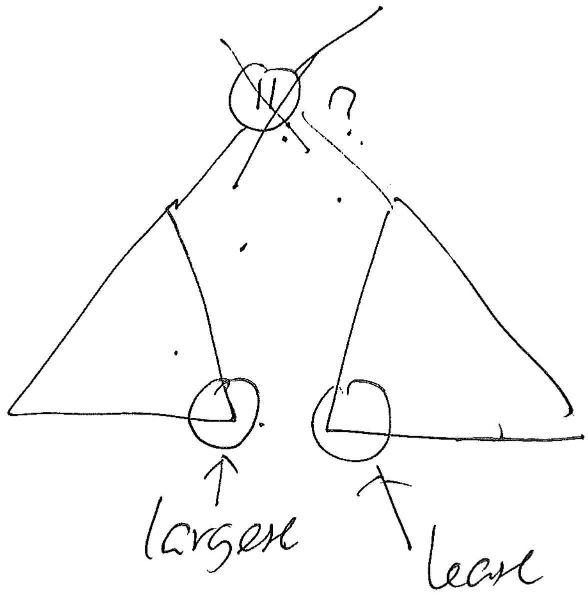
A binary tree is a BST if for any node, its is < all the keys in its right ~~stree~~ subtree

> all the keys in its left subtree

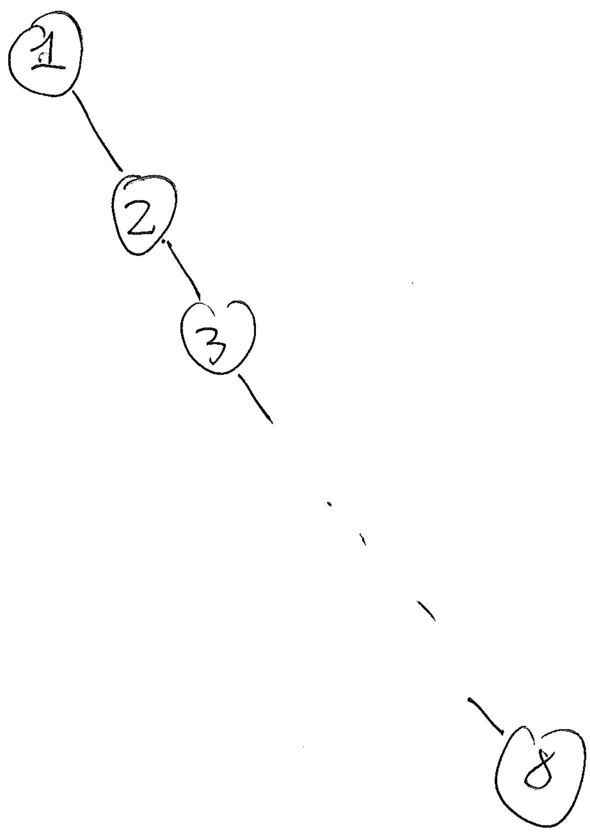
search(11)

insert(11)





insert 1, 2, 3, 4, 5, 6, 7, 8



$O(n)$ time for inserting deletion and search

The key to guarantee good performance for a BST is to make sure the height is $O(\log n)$

A BST is said to be balanced if its height is $O(\log n)$

AVL-tree.

Def. A BST is called an AVL-tree if for every node the height difference between its left and right subtrees is ≤ 1

Observation, the height of an AVL-tree is bounded by $2\log n$

Tight bound: $1.4404 \log(n+2) - 0.328$

Let h be the height of an AVL tree with n vertices.

$$h \leq 2 \log n$$

Pf (Induction on h) $\quad n \geq 2^{h/2}$

Basis $h = 0 \qquad n = 1$

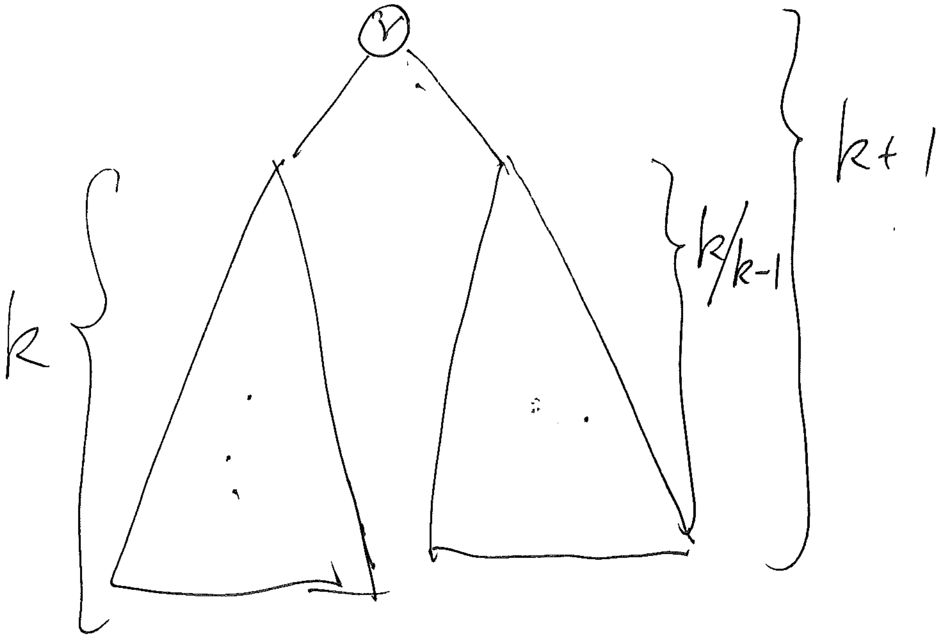
$$2^{h/2} = 2^{0/2} = 2^0 = 1$$

$$n = 1 \geq 1 = 2^{h/2}$$

I.S.

Assume that for all AVL-trees of height $h \leq k$ the claim is true

Need to show the claim is also correct for $\underline{h = k+1}$, i.e., for an AVL-tree of height $k+1$ there are at least $2^{\frac{k+1}{2}}$ nodes in the tree.



Case 1 left subtree has height k and right subtree has height k−1

$$\geq 1 + 2^{\frac{k}{2}} + 2^{\frac{k-1}{2}} \geq 2^{\frac{k}{2}} + 2^{\frac{k-1}{2}} \geq 2^{\frac{k-1}{2}} + 2^{\frac{k-1}{2}}$$

$$= 2^{1} \cdot 2^{\frac{k-1}{2}} = 2^{1+\frac{k-1}{2}} = 2^{\frac{2+k-1}{2}} = 2^{\frac{k+1}{2}}$$

Case 2 both subtrees have height k

$$\geq 1 + 2^{\frac{k}{2}} + \cancel{2^{\frac{k}{2}}}\ 2^{\frac{k}{2}} \geq 1 + 2^{\frac{k}{2}} + 2^{\frac{k-1}{2}} \geq 2^{\frac{k+1}{2}}$$

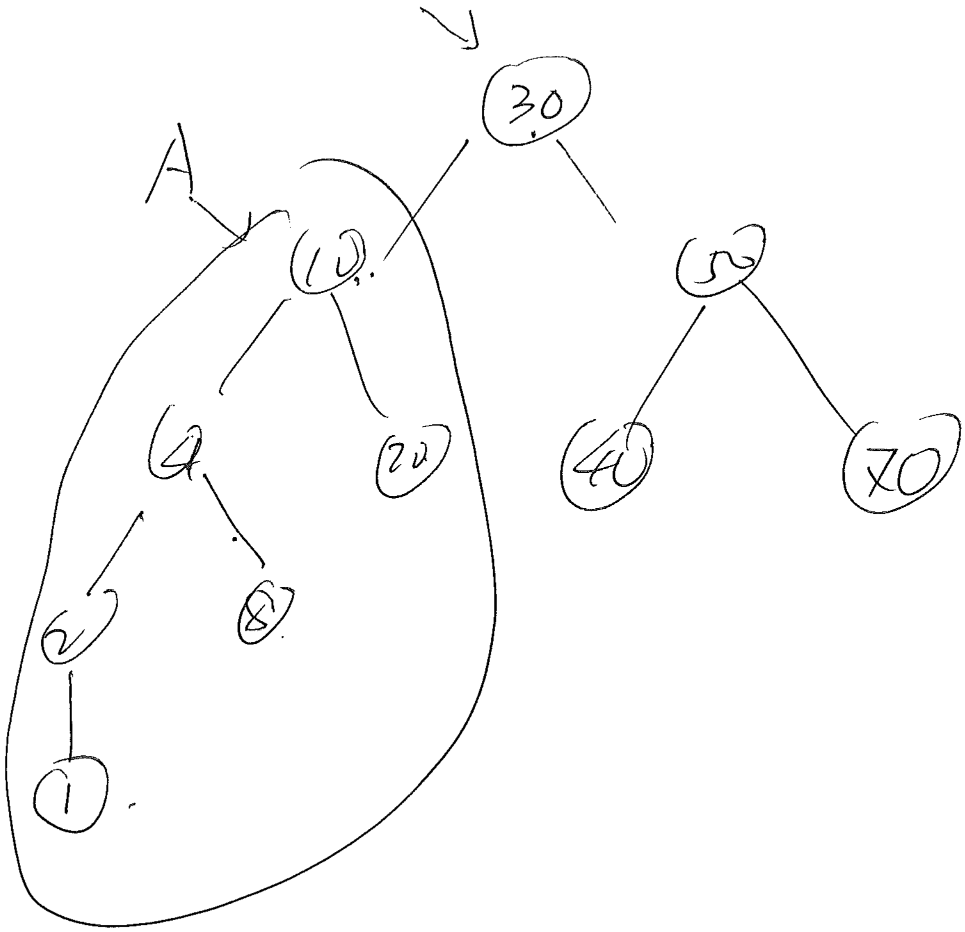
Search, insertion and deletion in an AVL-tree

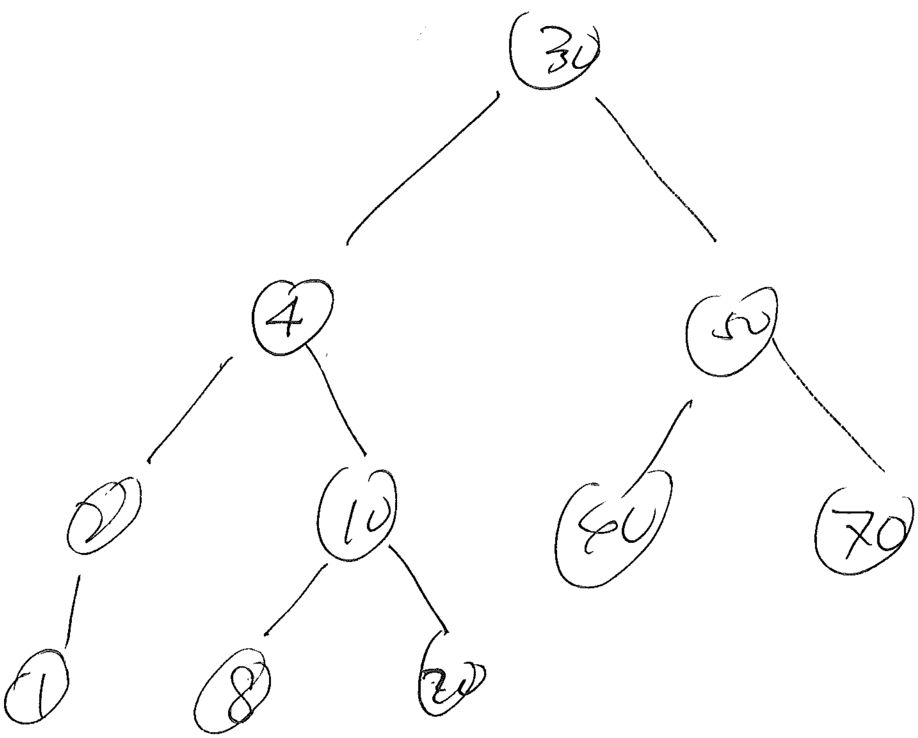
Search just like a normal BST.

insertion and deletion.

1. We will perform a normal insertion and deletion just like a BST
2. If the insertion or deletion violates the AVL property, then $\underbrace{\text{we rebalance the tree}}_{\text{rotations}}$

Let A be the ~~node~~ root of the smallest subtree that's violating the AVL property after insertion


A
30
10
50
4
20
40
70
2
8
1


30
4
50
2
10
40
70
1
8
20