CSE 101 3-5-25

Part: One more day: Sat 3/8

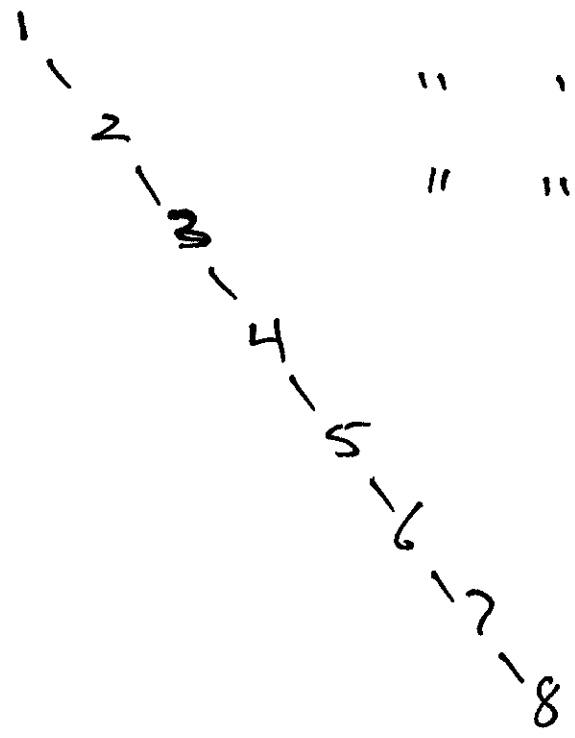
Ex. insert 1, 2, 3, 4, 5, 6, 7, 8

in order, we get

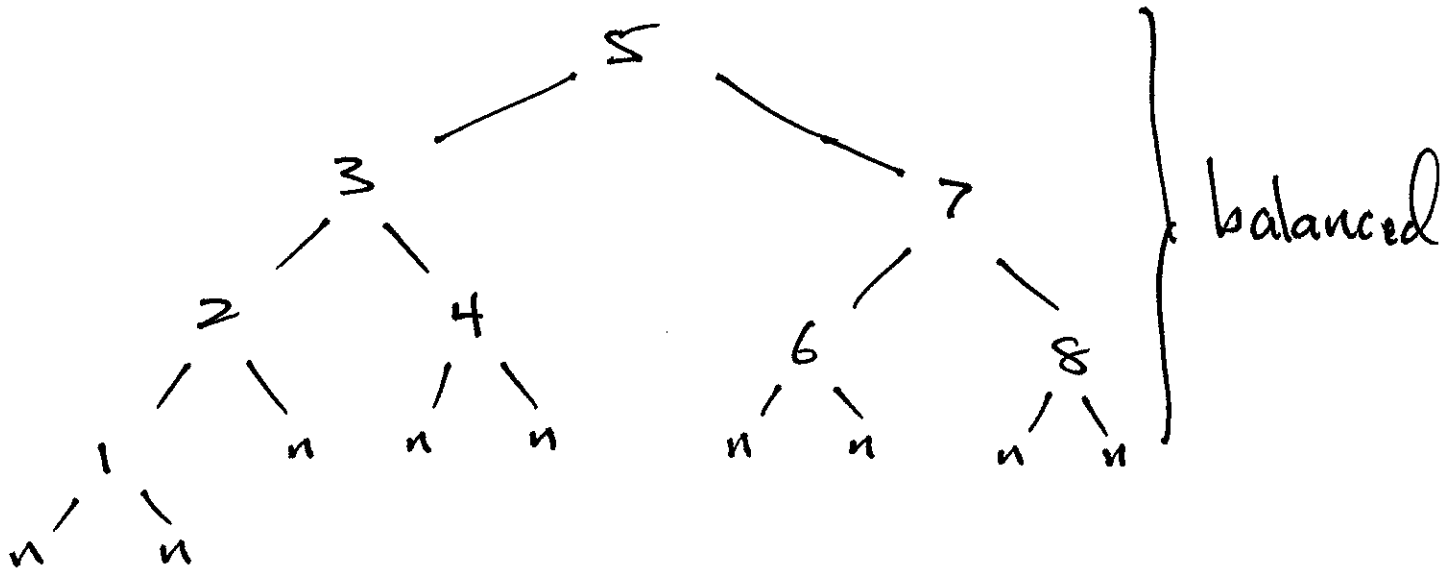
cost of search $= (n-1) = \Theta(n)$

" " insert "

" " delete "



Ex. insert: 5, 3, 2, 4, 1, 7, 6, 8



cost of search = $\Theta(\log n)$

" " insert "

" " delete "

To get a good Worst case runtime ($\Theta(\log n)$, where n = #keys) use

Red-Black trees (RBT)

RBT Properties

1. each node is either Red or Black.
2. root is Black.
3. each leaf (i.e. nil child) is Black

4. each Red node has 2 Black children, one or both of which could be nil.

equivalently, no descending Path has 2 Reds in a row.

5. for any node x, every descending Path from x to a leaf contains the same number of Black nodes.

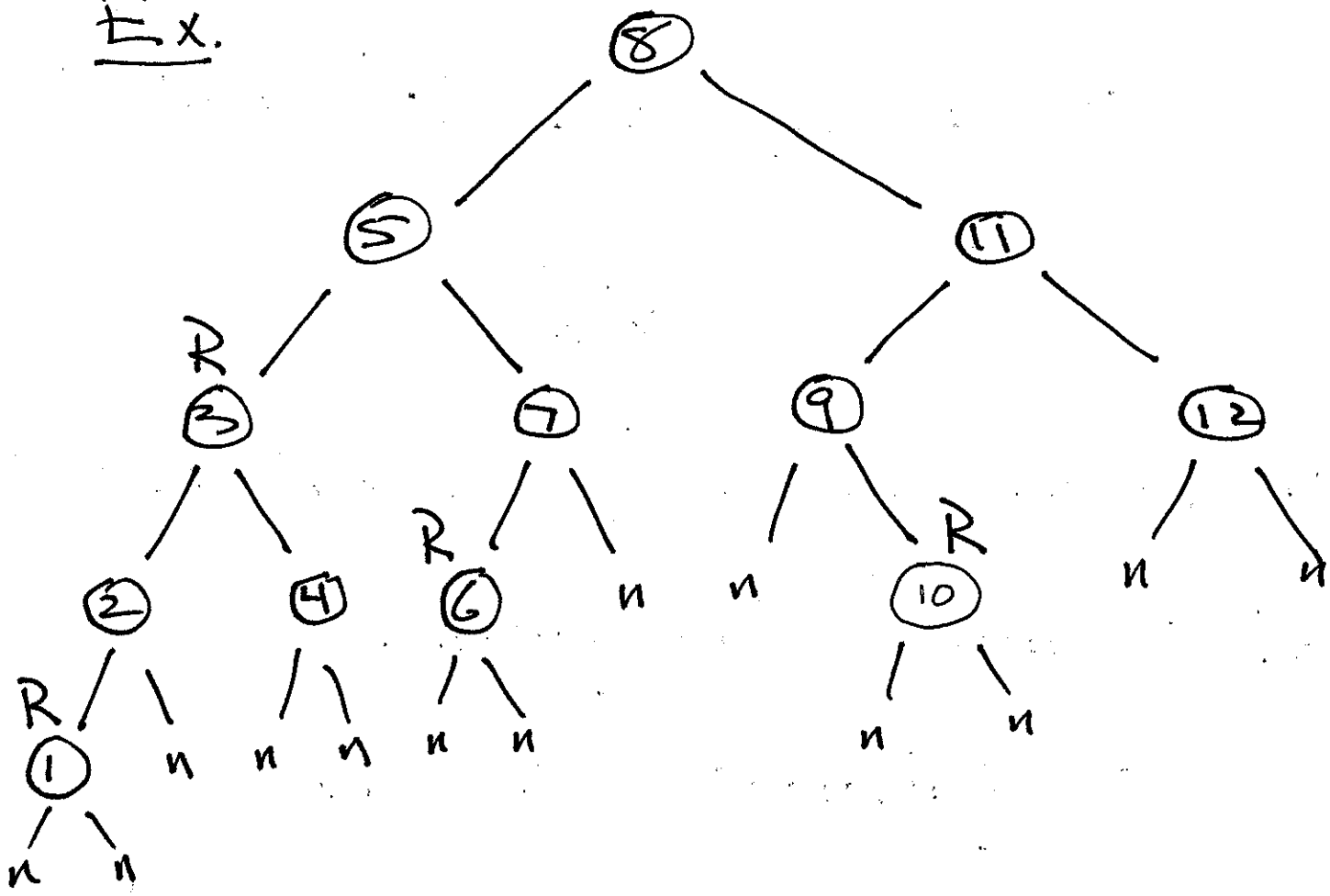
Defn

The Black height of x

$bh(x)$

is the # of black nodes in a descending Path from x to a leaf, not counting x.

Ex.



black height	node
3	8
2	5, 11, 3
1	1, 2, 4, 6, 7, 9, 10, 12
0	all nils.

Note:

for any x

$bh(x) = 0$ iff $height(x) = 0$

iff x is a leaf

Theorem

A RBT with n internal (i.e. non-nil) nodes, and height h satisfies

$$h \leq 2 \lg(n+1)$$

Rmks

- any Binary Tree satisfies

$$h \geq \lfloor \lg n \rfloor$$ { will show later

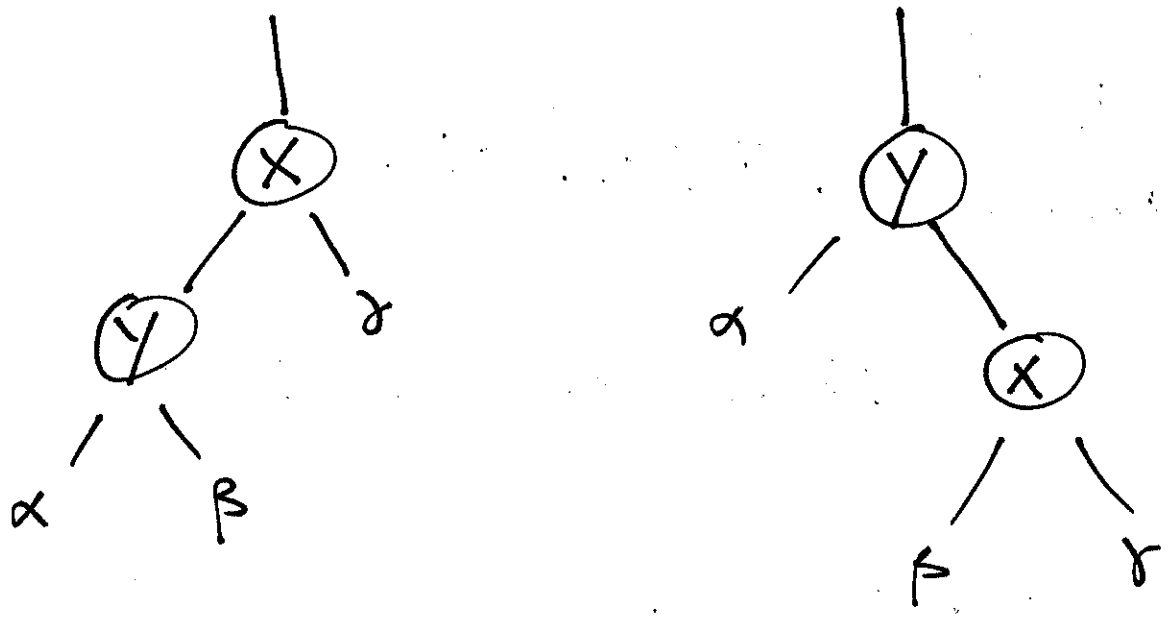
- so a RBT satisfies

$$\underbrace{\lfloor \lg n \rfloor}_{\Omega(\log n)} \leq h \leq \underbrace{2 \lg (n+1)}_{O(\log n)}$$

- Thus $h = \Theta(\log n)$

- Therefore ... BST ops. have cost $\Theta(\log n)$

13.2 Rotations



RightRotate(T, x)

LeftRotate(T, y)