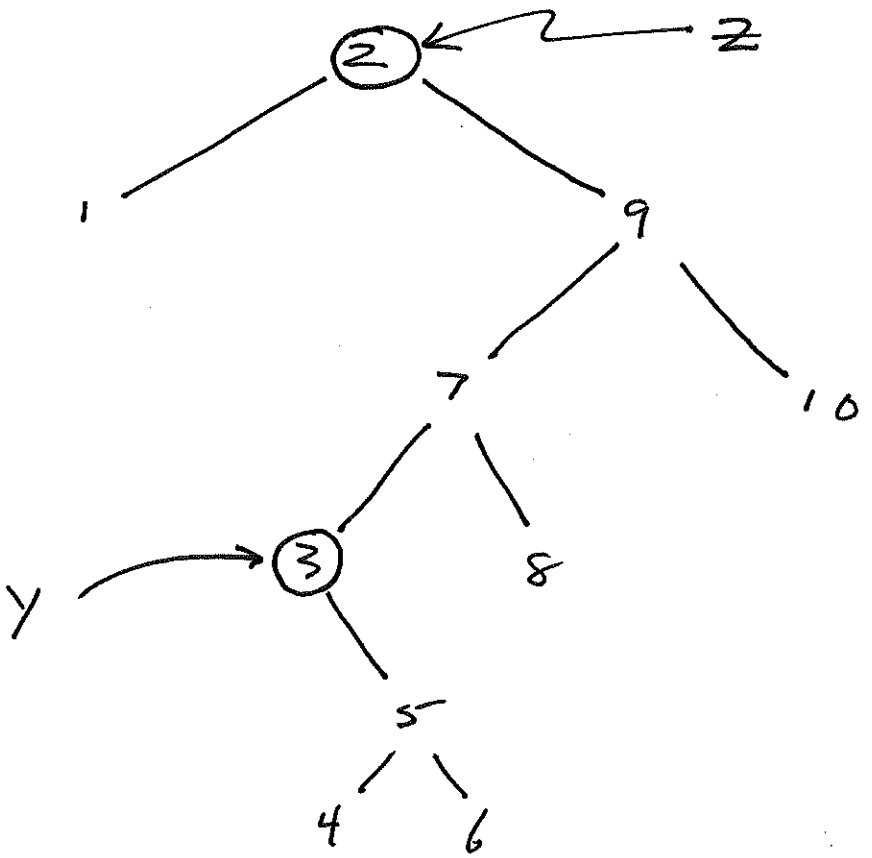
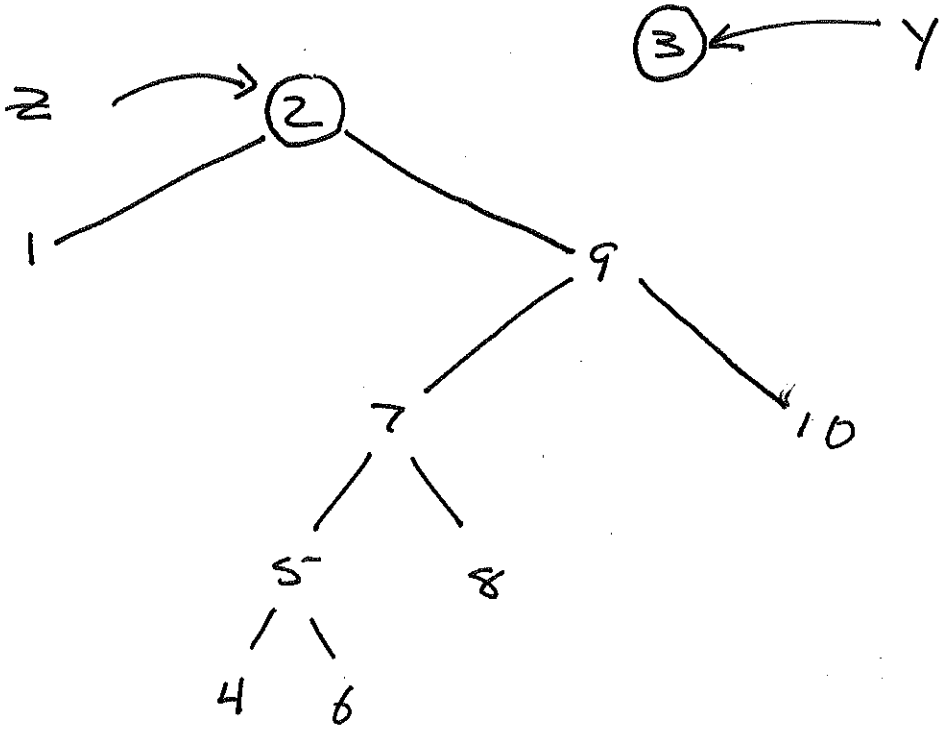
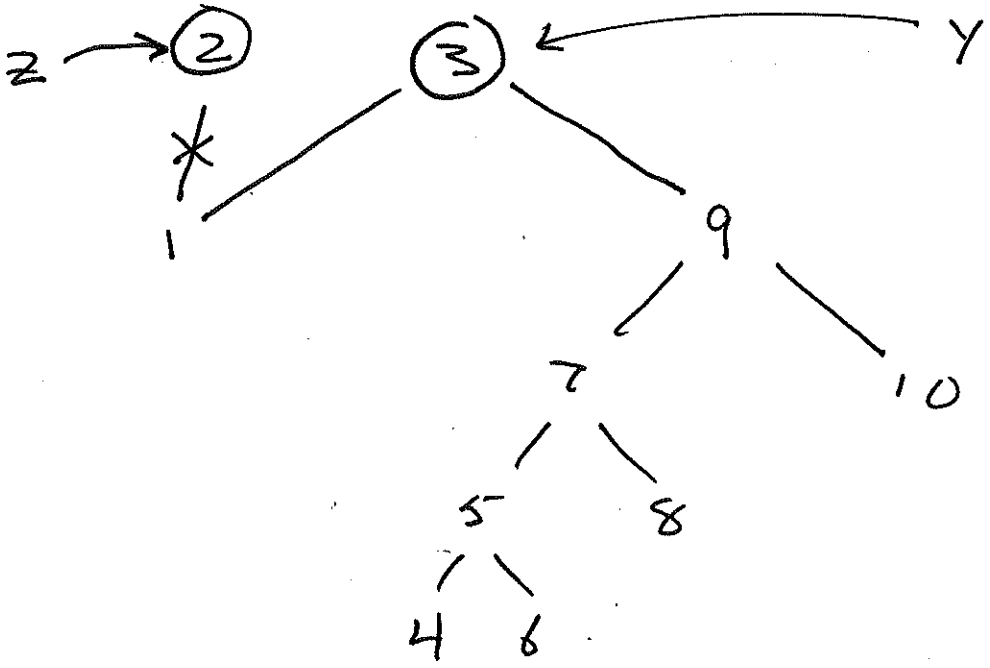
Par: ext. 1 day → Thur.

Pas: Posted due mon 6/5.

Ex Delete(T, z) (case 3)




Z
2
3
Y
1
9
7
10
5
8
4
6


Z
2
3
Y
1
9
7
10
5
8
4
6

Pa7:

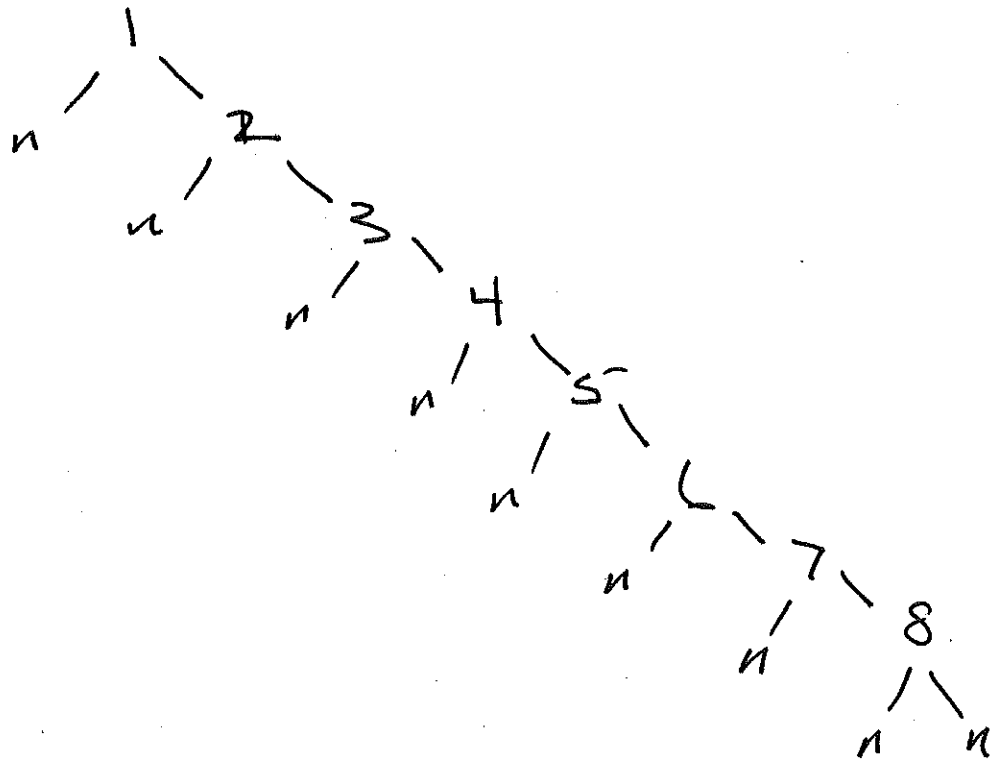
- forward & backward iteration

analogy

PairList	Dictionary
moveFront()	⟷ begin()
moveBack()	⟷ end()
get()	⟷ currentKey() / currentVal()
moveNext()	⟷ next()
movePrev()	⟷ prev()

Problem with BSTs:

Ex. insert 1 2 3 4 5 6 7 8



Runtime: queries, insert, lookup, delete are all

$$\Theta(\text{height}(T))$$

Solution: Red Black Trees (RBT)

(chap 13 in CLRS)

Defn An RBT is a BST that also satisfies RBT properties.

Convention: the leaves in a RBT are the nil children of key-bearing nodes.

RBT Properties:

1.) Each node is Red or Black.

2.) root is Black.

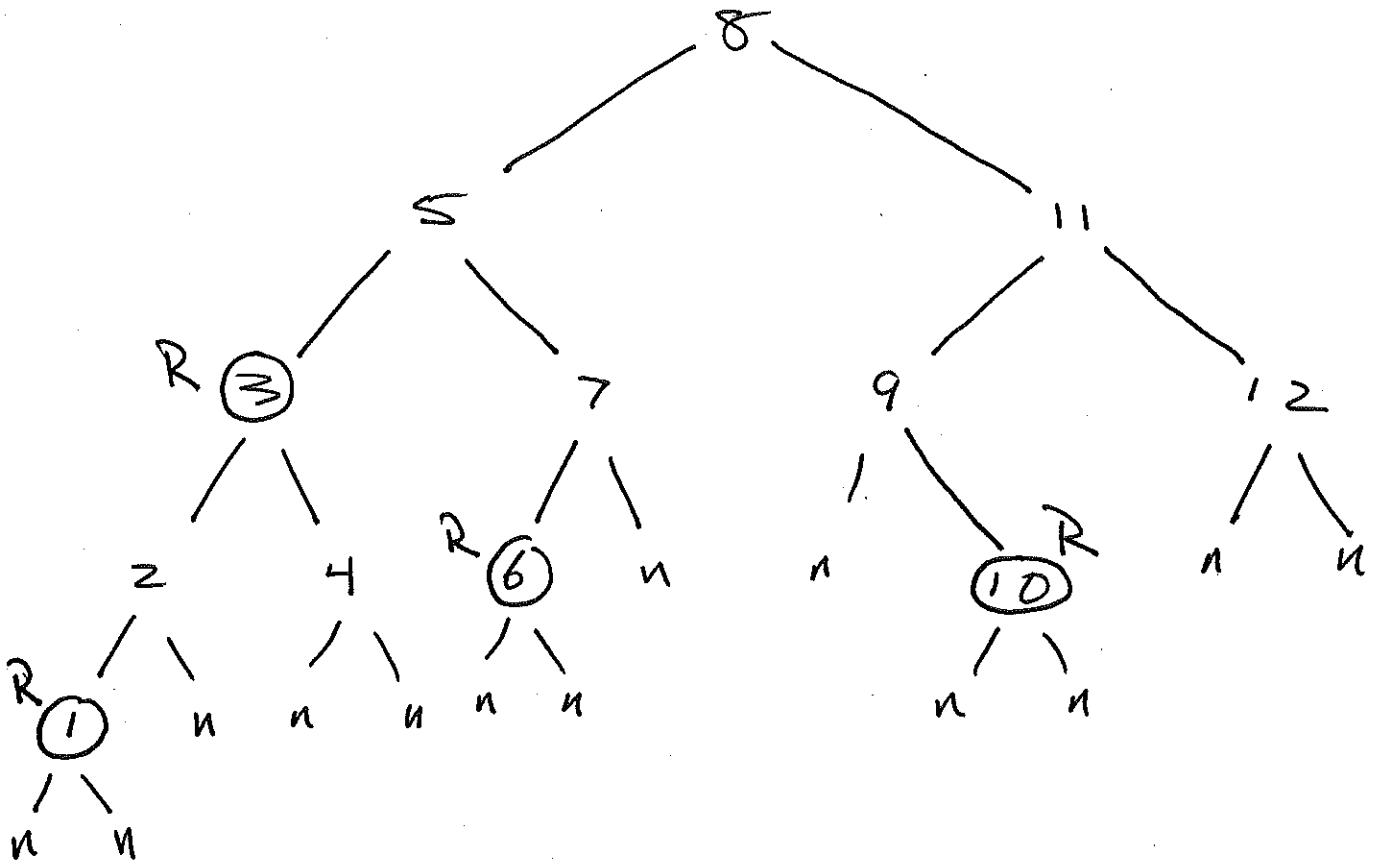
3.) Each leaf (nil child) is Black.

4.) Every Red node has 2 Black children (possibly nil).

5.) For any node x in a RBT, every descending Path from x to a descendant leaf has the same number of Black nodes.

Defn The black-height of a node x is the # of black nodes in a desc. Path from x to a leaf, not counting x itself.

Ex.



node	black-height
8	3
5, 11, 3	2
1, 2, 4, 6, 7, 9, 10, 12	1
all nil leaves	0

note: $bh(x) = 0$ iff $height(x) = 0$ iff x is a leaf.

Theorem

A RBT with n internal (key-bearing) nodes and height h satisfies

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

Rmks

- any Binary Tree satisfies

$$h \geq \lfloor \lg n \rfloor$$

Thus a RBT T satisfies

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

$\Rightarrow$ $\text{height}(T) = \Theta(\log n)$. "Balanced"