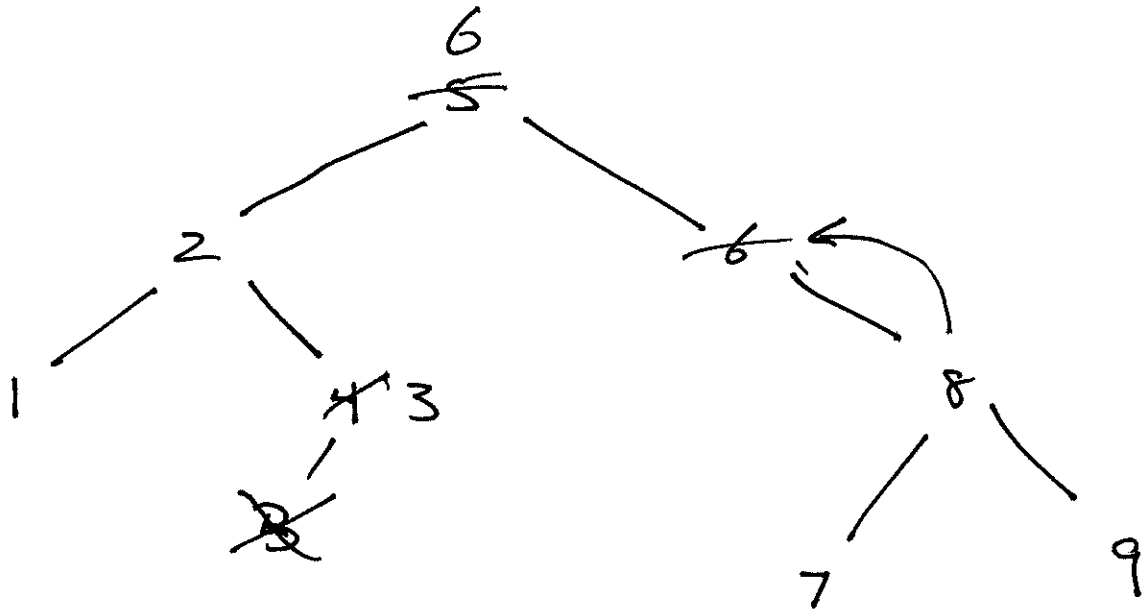


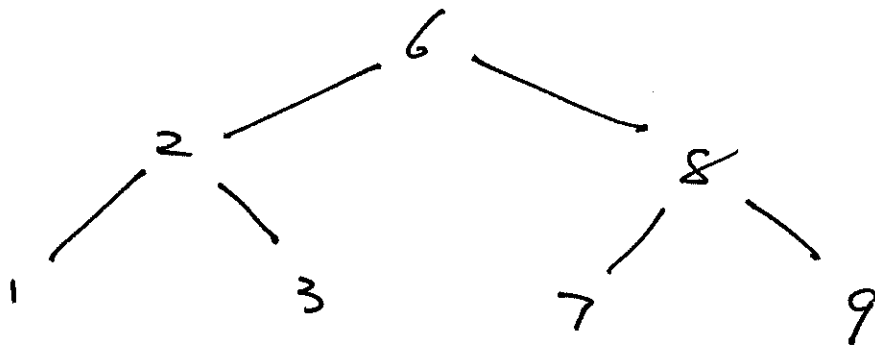
CSE 101 3-5-24



EX insert: 5 2 6 4 3 8 7 1 9

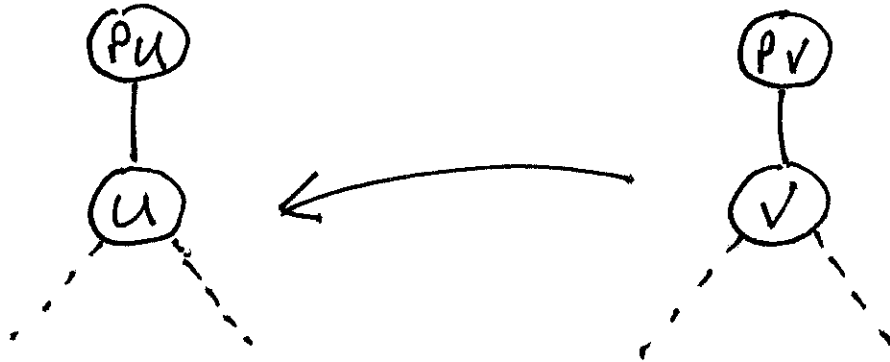


Delete: 4, 5

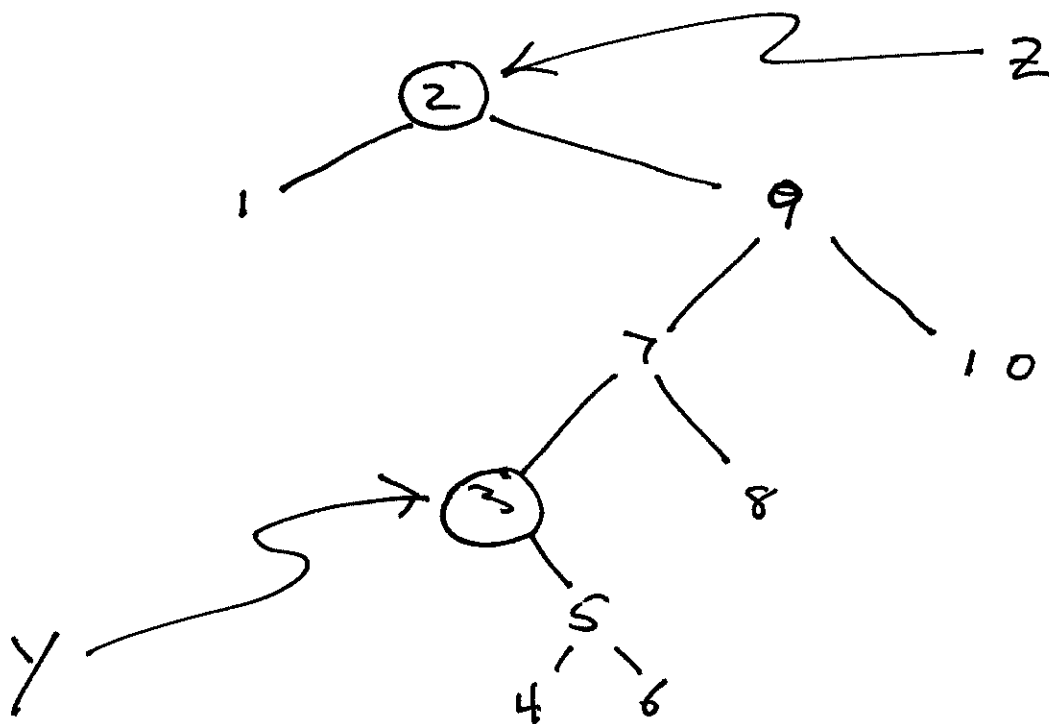


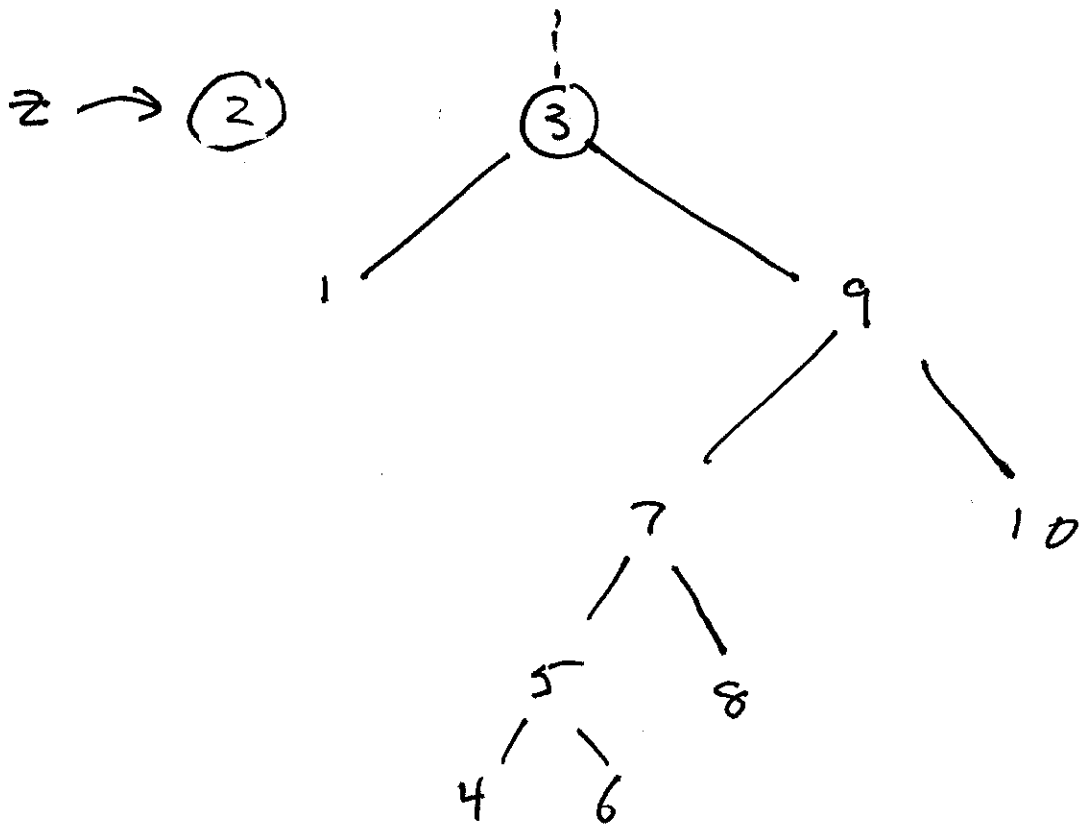
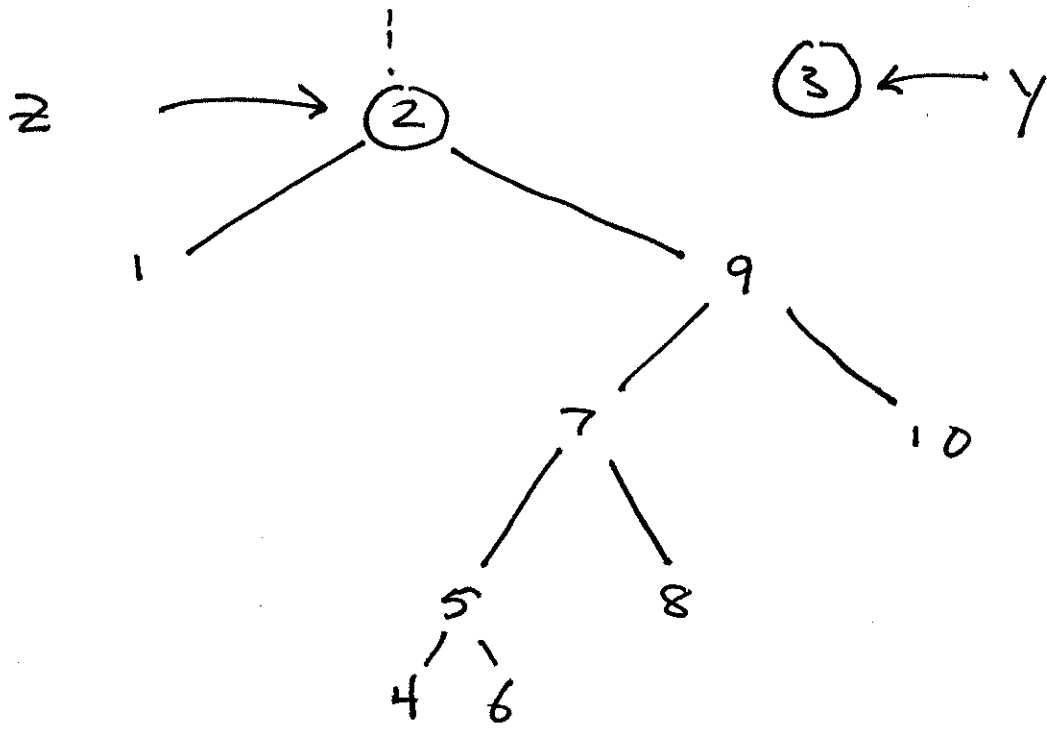
# Pseudo-code for Delete

• Transplant ( $T, u, v$ )



Ex. Delete ( $T, z$ ) (case 3)





PaT: forward & backward iteration

Analogy:

list in PaT

moveFront()

moveBack()

get()

moveNext()

movePrev()

Dictionary in PaT

begin()

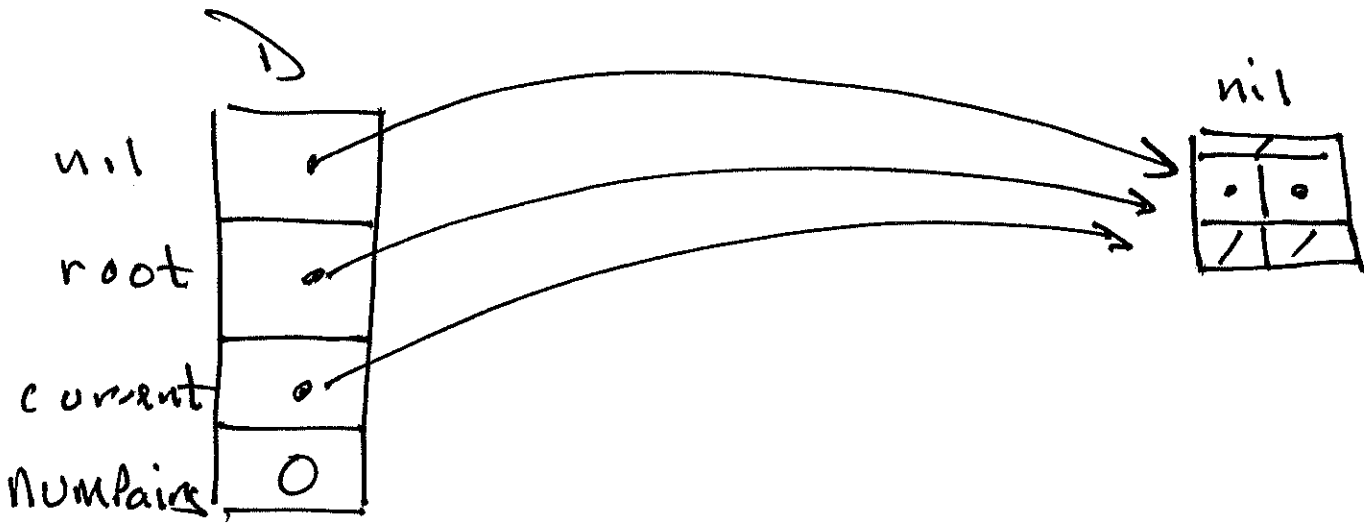
end()

{ currentKey()  
currentVal()

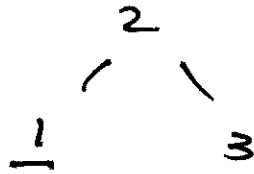
next()

prev()

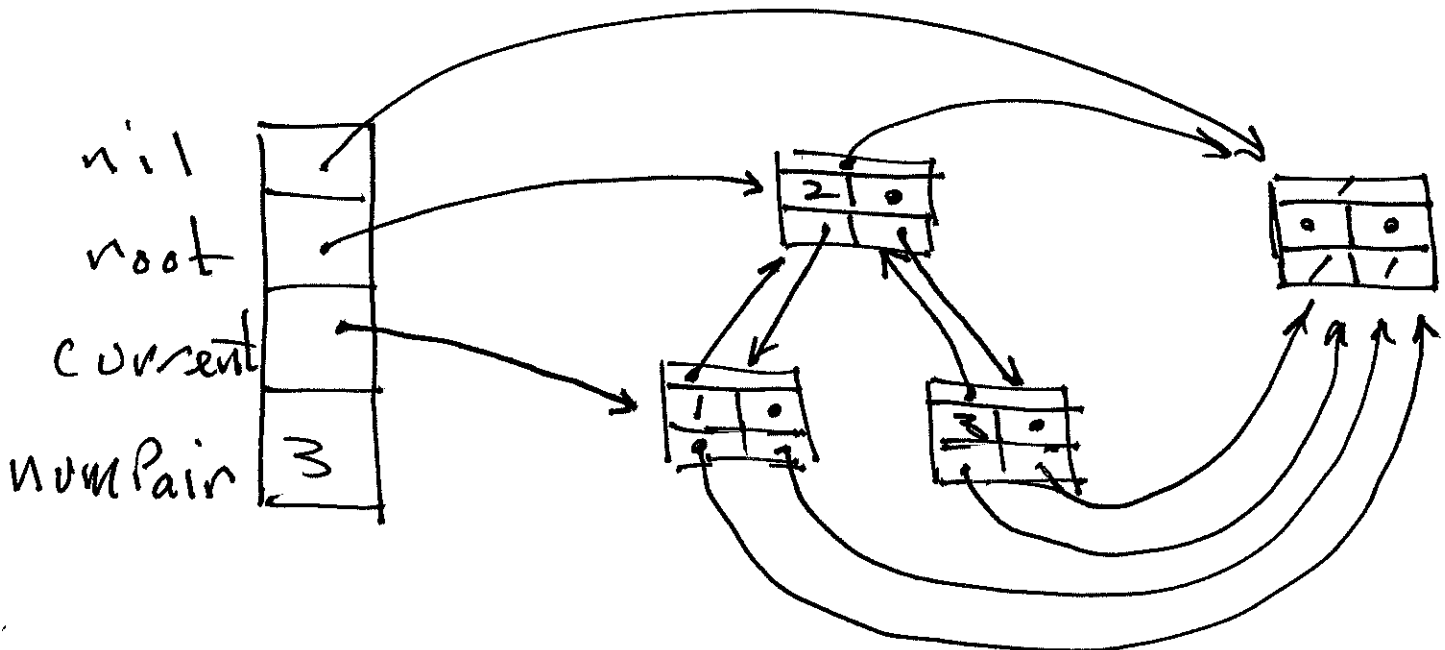
# Dictionary empty state



client view:

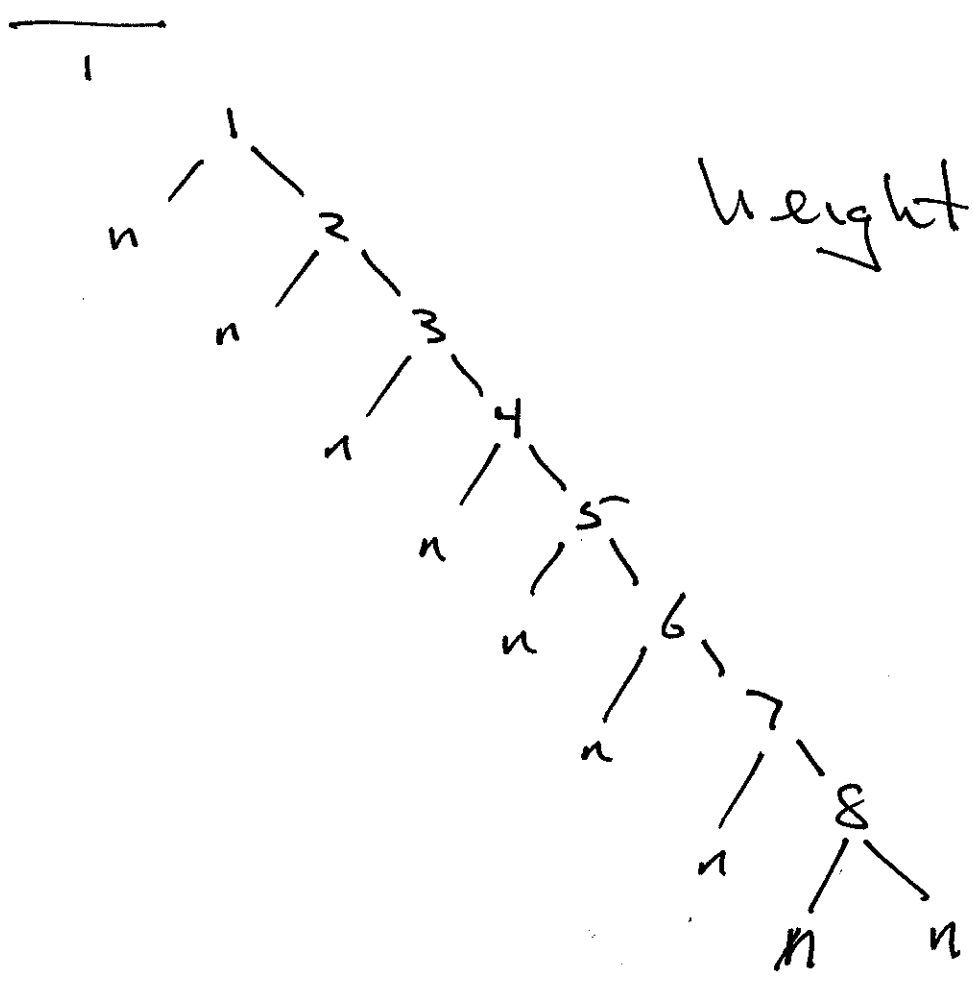


inside view:



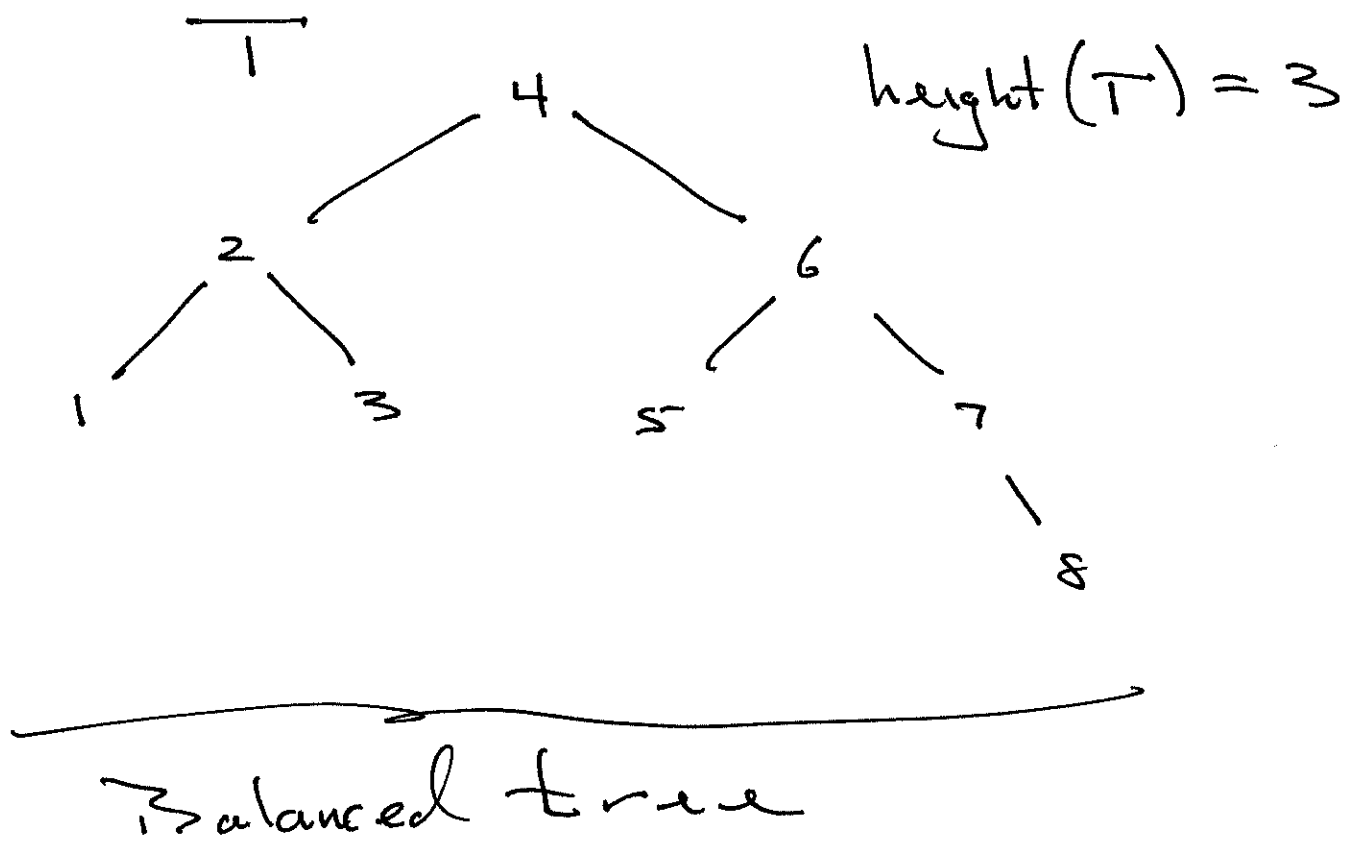
what's wrong with BSTs ?

Ex insert: 1 2 3 4 5 6 7 8



height(T) = 7

Ex. better!   1   2   6   1   3   5   7   8



Runtime of queries, insert, delete:

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

For any BST on  $n$  nodes

$$\lfloor \lg n \rfloor \leq \text{height}(T) \leq n-1$$

Balanced

Solution : Red-Black Trees (RBT)

chapter 13.

### 13. RBTs

convention: leaves in a RBT will be the nil children of key-bearing nodes.

Defn

A RBT is a BST that in addition to the BST Properties, satisfies the RBT Properties.

## RBT Properties

1. each node has color Red or Black.
2. root is Black
3. each leaf (i.e. nil) is Black.
4. every Red node has 2 Black children (one or both may be nil.)
5. for any node  $x$ , every descending path from  $x$  to a leaf (nil) contains same # of black nodes.

Defn

The Black Height of  $x$ , denoted  $bh(x)$ , is the # of black nodes in any descending path from  $x$  to a leaf (nil), not counting  $x$ .