CSE 101 11-19-20

Pas: ext 2 days

Pas: question

cleanUp() :

client view :

① + ② 2 | 2 2 ③ 3

Quiz 4 : next Tue.

Pa6: Posted

Runtimes:

- Traversals: InOrderTreeWalk(x)
  Pre
  Post
  (x: root[T])

cost: $\Theta(n)$ where n = #nodes in T.

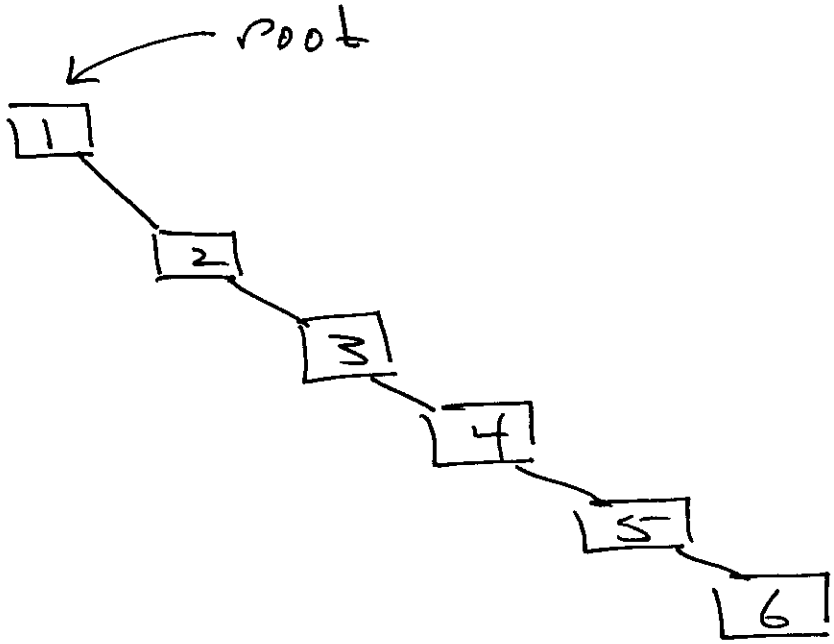
- Queries: TreeSearch(x)
  .. Min(x)
  .. Max(x)
  .. Successor(x)
  .. Predecessor(x)

Defn height(T) = dist. from root to deepest leaf.

Defn height(x) = height of subtree rooted at x.

worst case cost: Θ(height(x))
or Θ(height(successor(x)))

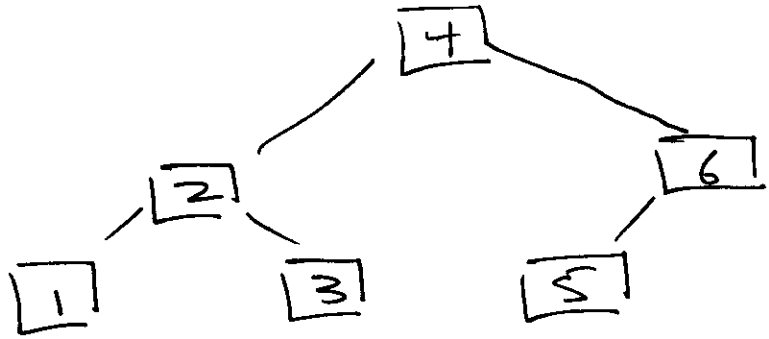
worst possible BST for Queries.



every node has only one child.

cost of query : $\Theta(n) = 6$

Best Possible BST :



cost of query : $\Theta(\text{height}(T)) = 2$

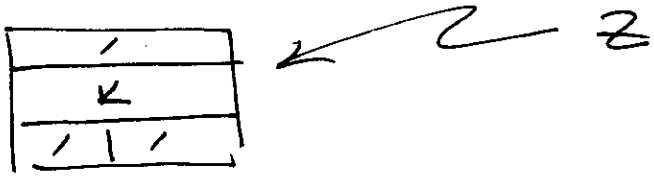
general: $\Theta(\log n)$

12.3 Insertion & Deletion

new node z

To insert z and maintain BST properties

- set key[z] = k
- set p[z] = left[z] = right[z] = nil.



- find a node to adopt z as left or right child.

strategy:

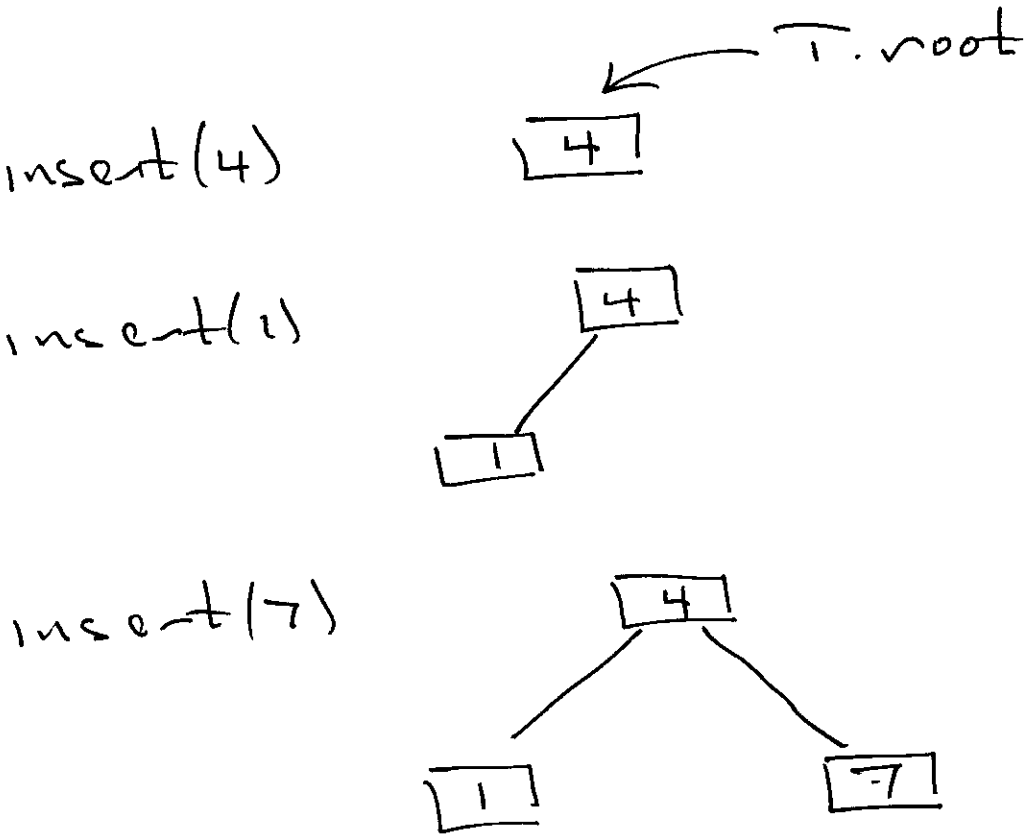
Do a search for key k in T, find nil child where k would reside, if it were in T.

Ex. TreeInsert(T, z)

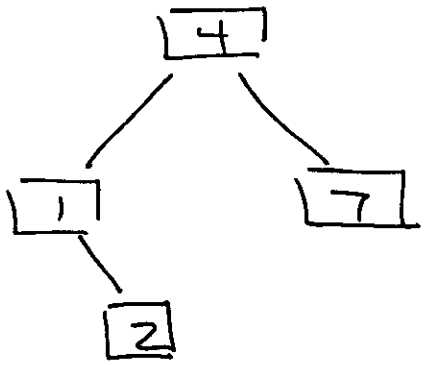
see pseudo-code

Keys: 4 1 7 2 8 3 5 6

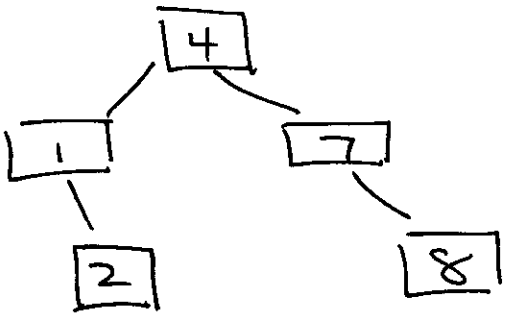
T.root = nil : empty state



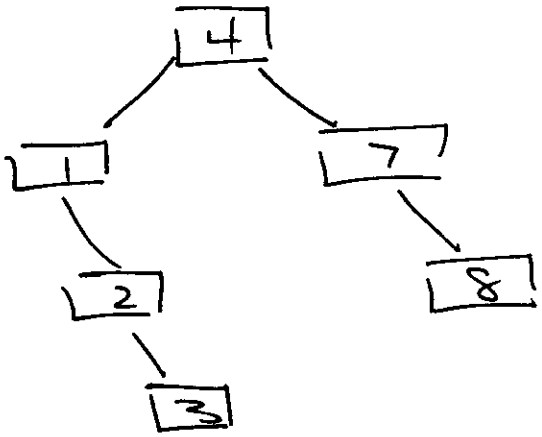
insert(2)



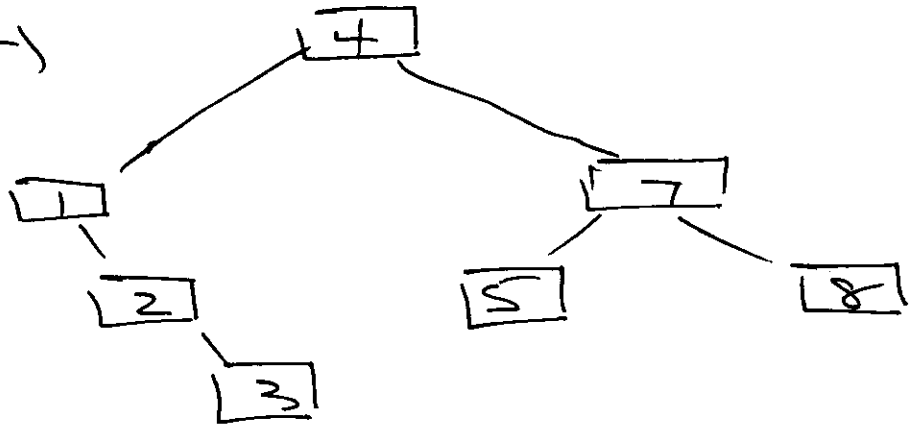
insert(8)



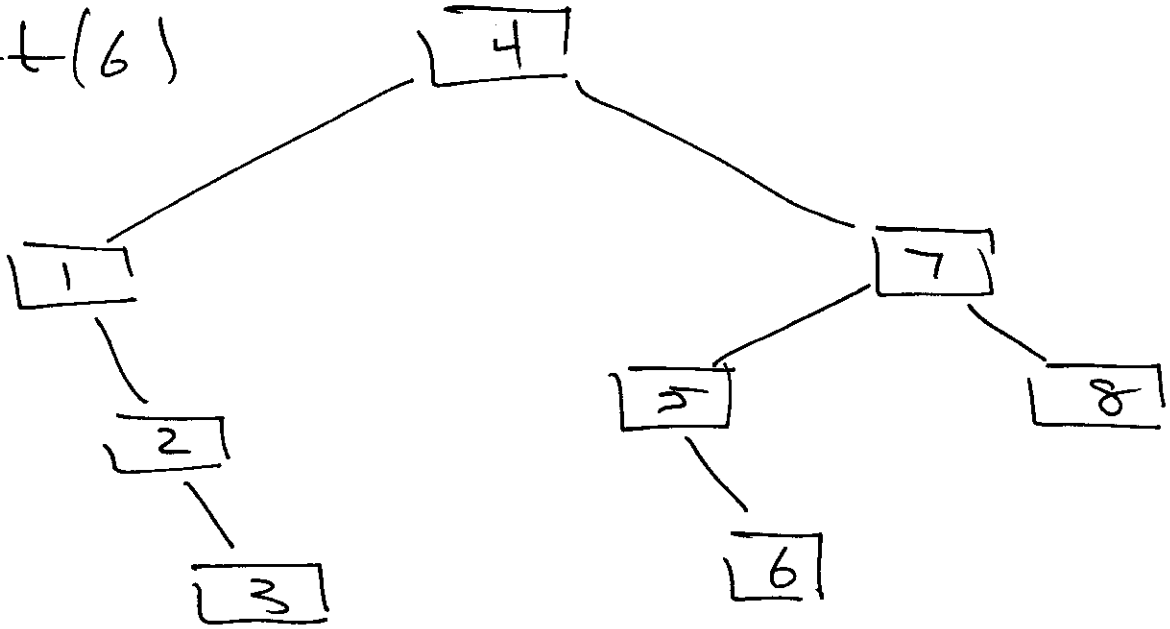
insert(3)



insert(5)



insert(6)



Runtime:

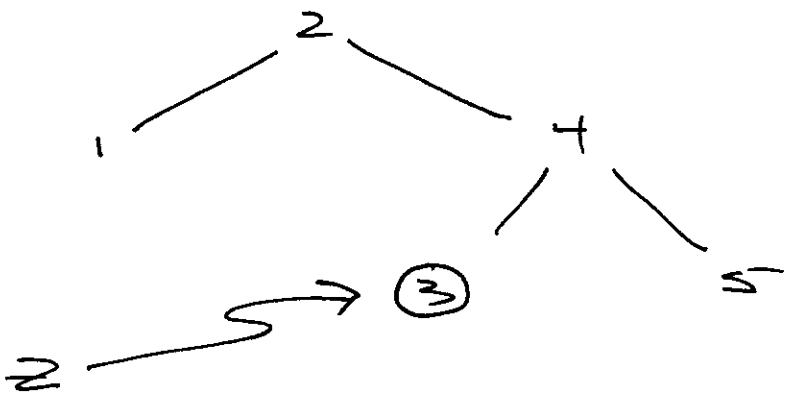
worst case cost : $\Theta(\text{height}(T))$.

Deletion : delete z from T

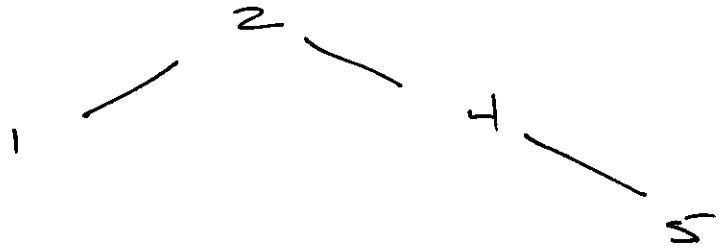
3 cases :

Case 1 : z is a leaf

detach z from its Parent



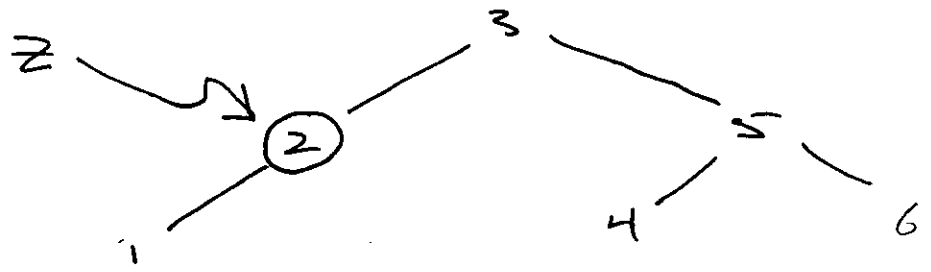
delete(3)



Case 2: z has 1 child.

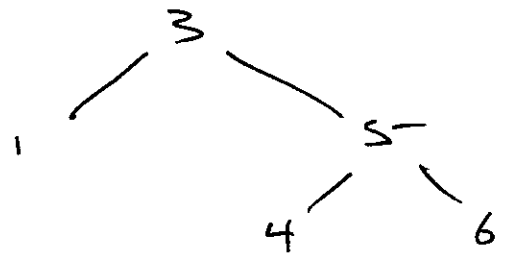
subcase 2.1 z has right but no left

subcase 2.2 z has left but no right



splice z out of 'local' linked list

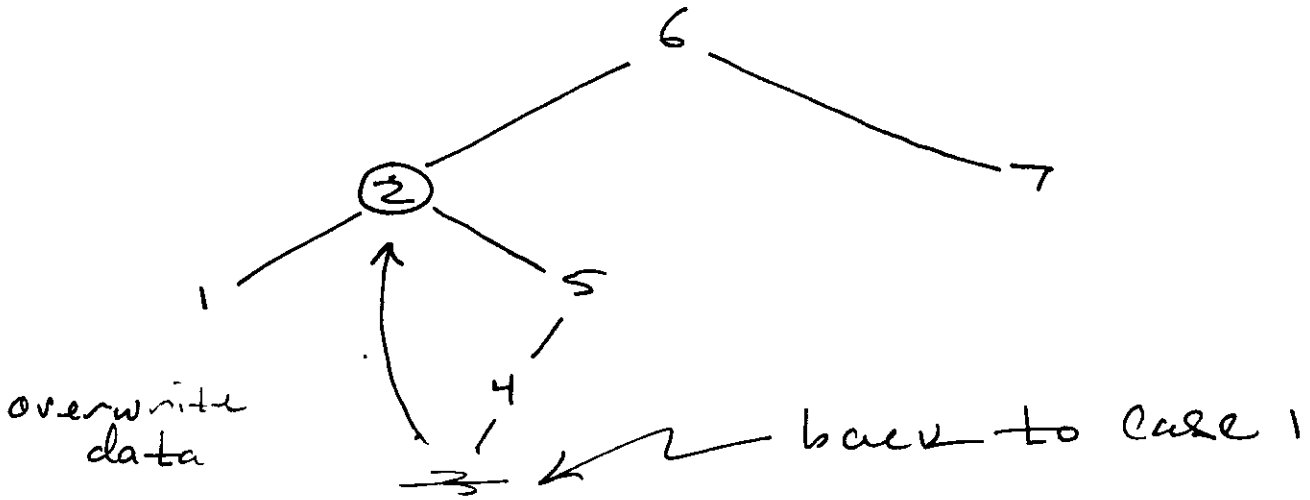
delete(z):
(Case 2.2)



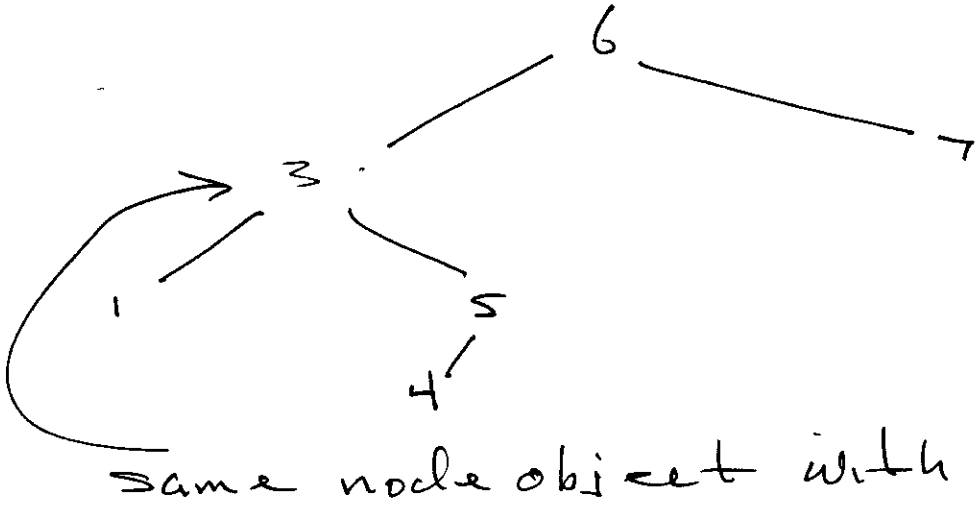
Exercise: draw example of subcase 2.1

Case 3: z has 2 children.

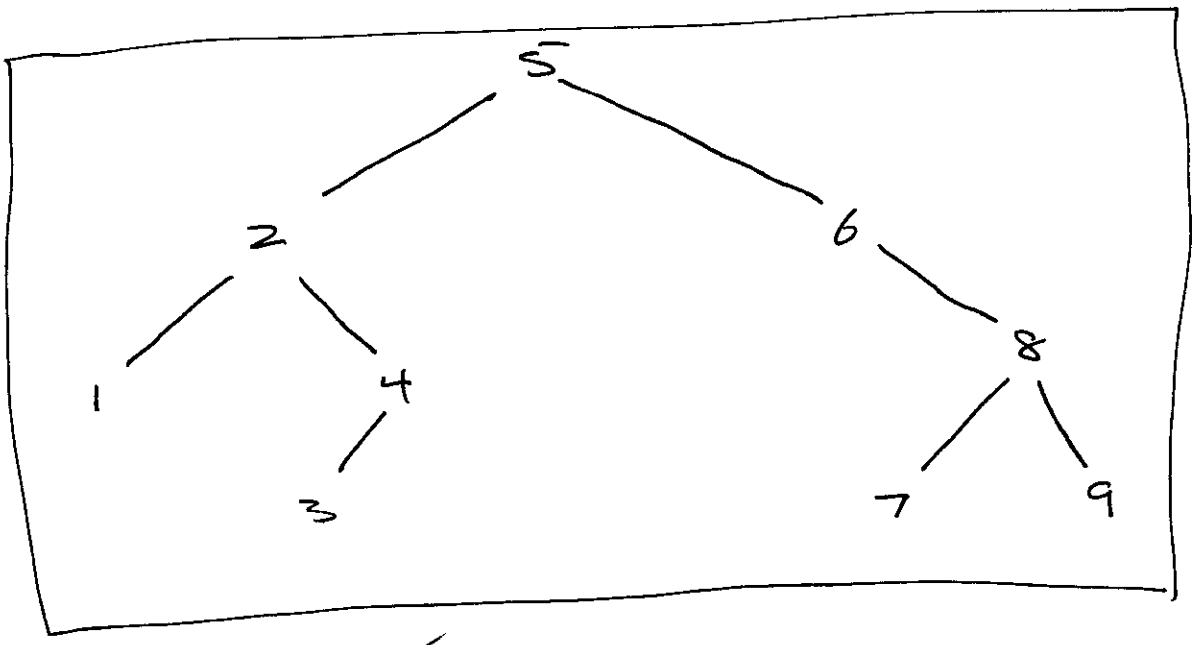
splice out z's successor y (which has no left child), then copy data of y into z.



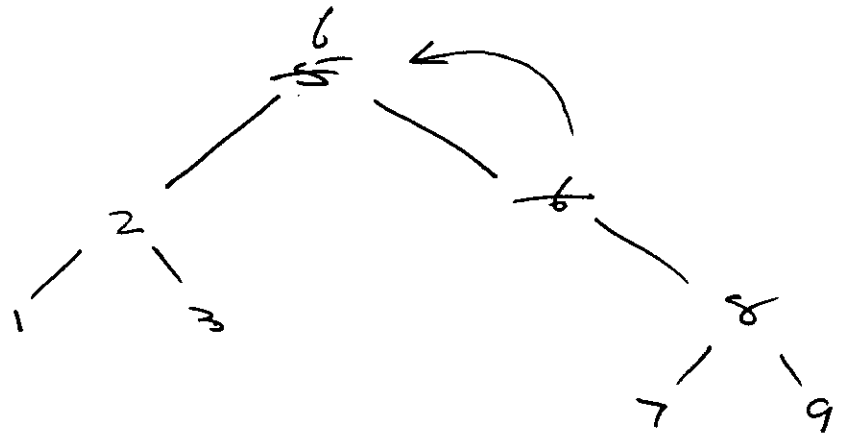
deleted(2)

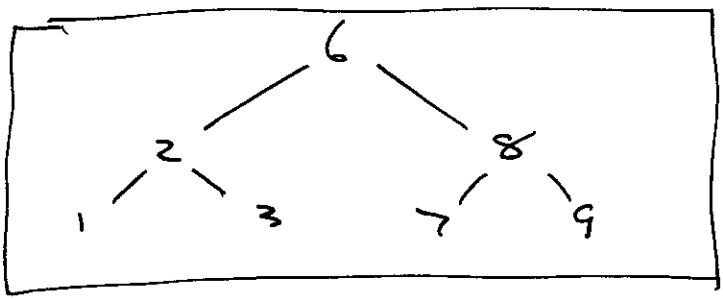


Ex. insert 5 2 6 4 3 8 7 1 9



Delete: 4, 5





Pseudo-code: next Time.