

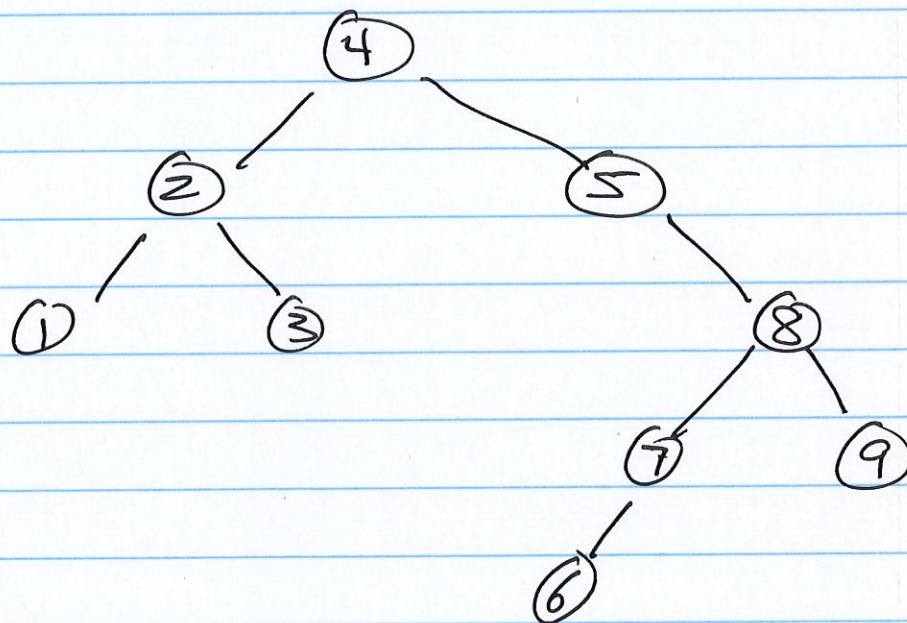
CSE 101 2-22-22

• Pa6 ext: 2 more days (last)
now due Friday.

• Quiz tonight

• Pa7 posted

Ex.



In: 1 2 3 4 5 6 7 8 9
Pre: 4 2 1 3 5 8 7 6 9
Post: 1 3 2 6 7 9 8 5 4

Queries

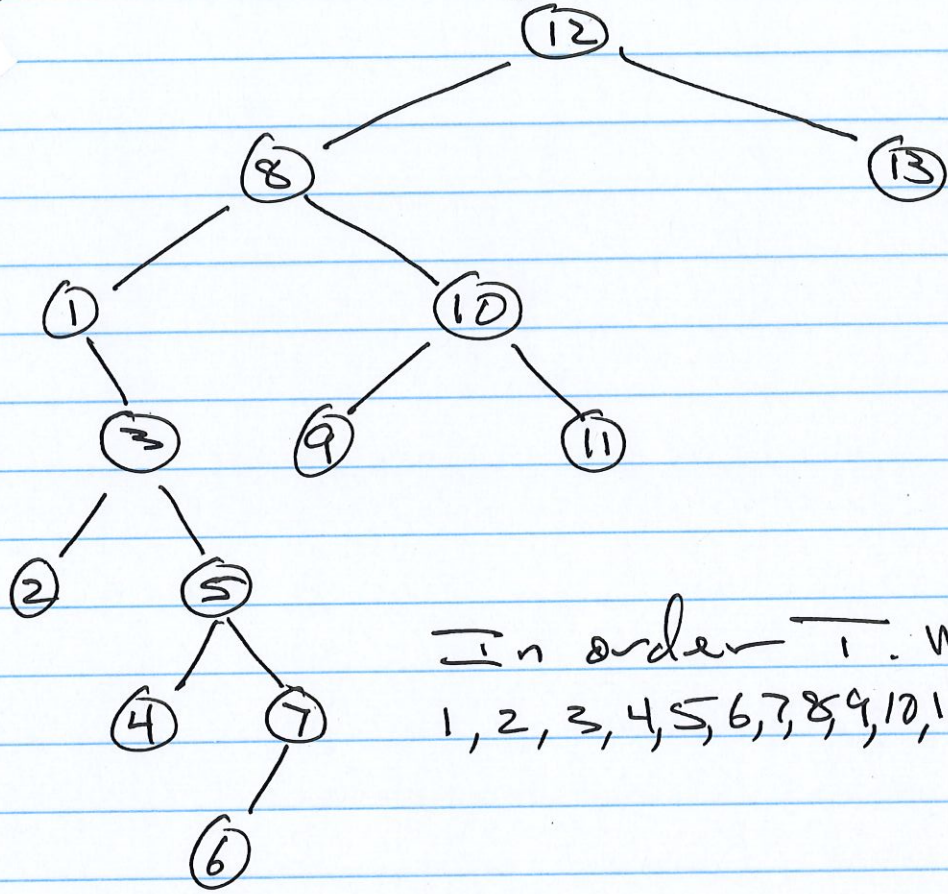
- $\text{TreeSearch}(x, k)$
- $\text{TreeMinimum}(x)$
- $\text{TreeMaximum}(x)$
- $\text{TreeSuccessor}(x)$
- $\text{TreePredecessor}(x)$

Defn

The Successor of Node x is the next node after x to be processed in an in-order tree-walk

Ex.

root →



In order T.W.:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Successor(1) = 2 (case 1)

Successor(8) = 9 (case 1)

Successor(7) = 8 (case 2)

Successor(13) = nil (case 2)

case 1 x has a right child.

Successor of x is leftmost node in x 's right subtree.

case 2 x has no right child.

Successor of x is the lowest ancestor of x whose left child is also an ancestor of x .

Exercise

Do all this for Predecessor.

Runtime (worst case)

• Traversal

In ord tree walk (x) ↖ root
 Pre " " "
 Post " " "

cost = $\Theta(n)$ where $n = \# \text{ nodes}$.

• Queries :

Tree Search(x)	} cost	$\Theta(\text{height}(x))$
• Min (x)		or
• Max		$\Theta(\text{height}(\text{succ}(x)))$
• Successor :		<u>one case</u>
• Predecessor(x)		of successor

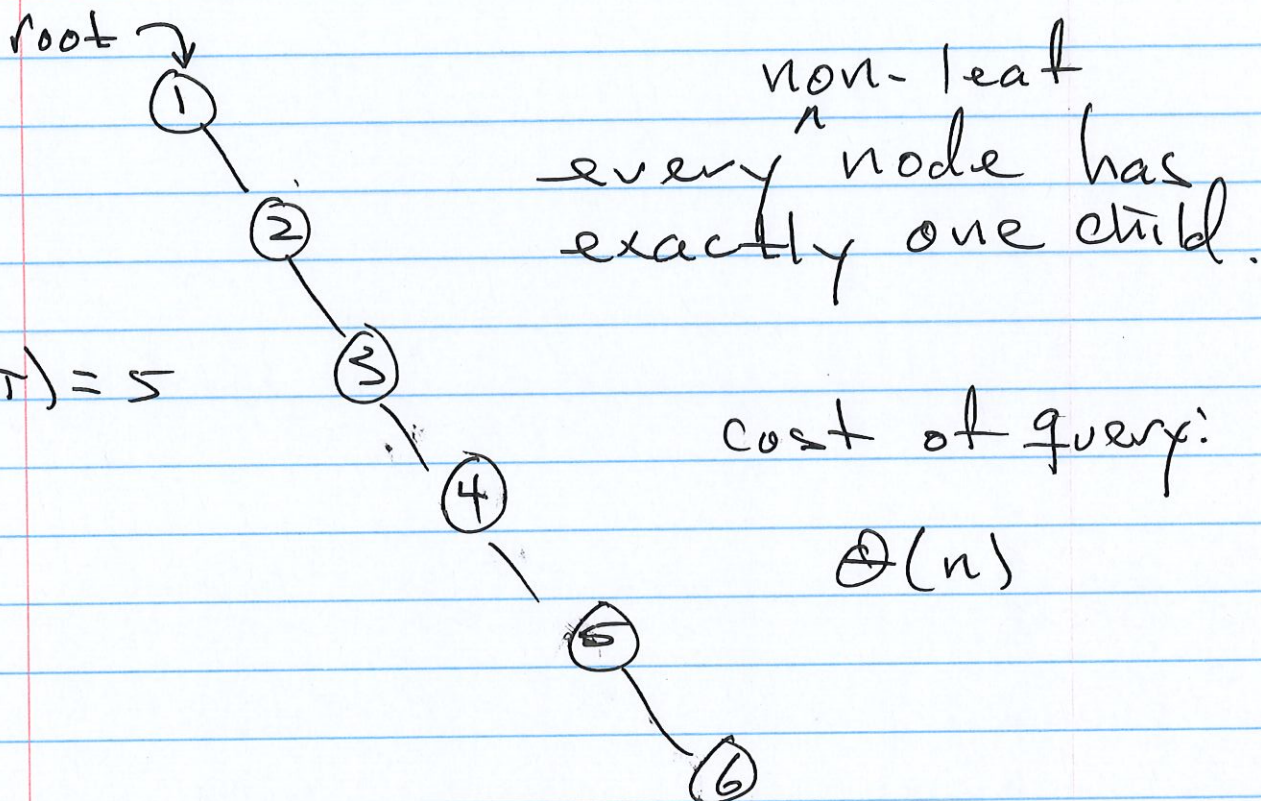
Defn

$\text{height}(T) = \text{dist. (depth) from root to deepest leaf.}$

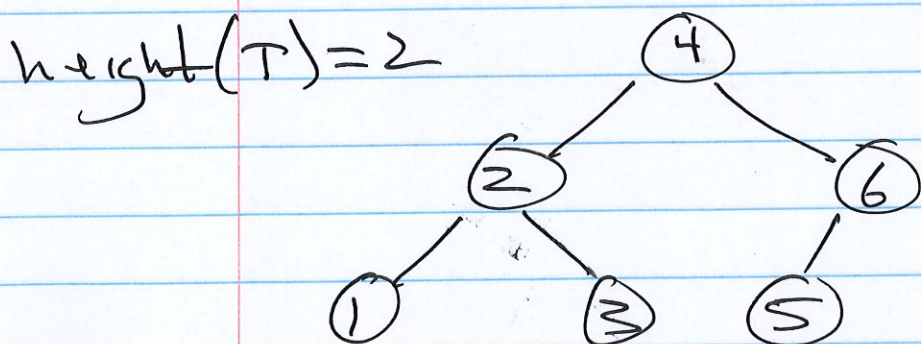
Defn

$\text{height}(x) = \text{height of subtree rooted at } x.$

Worst Poss. BST for Queries



Best Poss. BST for queries



□

In general

$$\Theta(\log n) \leq \text{height}(T) \leq \Theta(n)$$

"
(n-1)

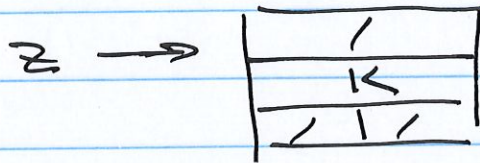
when $\text{height}(T) \approx \Theta(\log n)$,

we call T a balanced tree.

(12.3) Insertion & Deletion

To insert a new node z and maintain BST Properties:

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

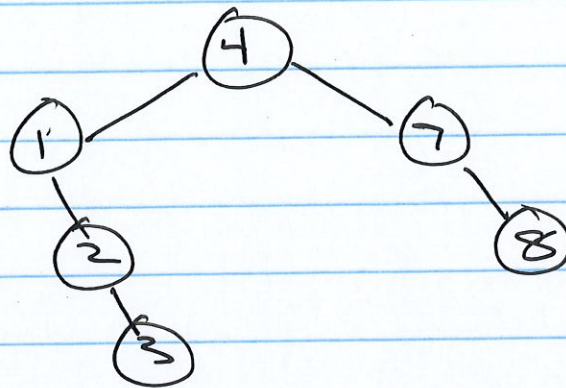


- find a node to adopt z as a child (left or right)

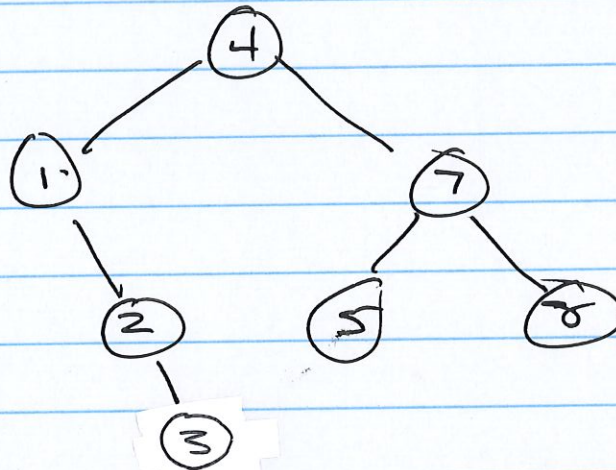
Strategy

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

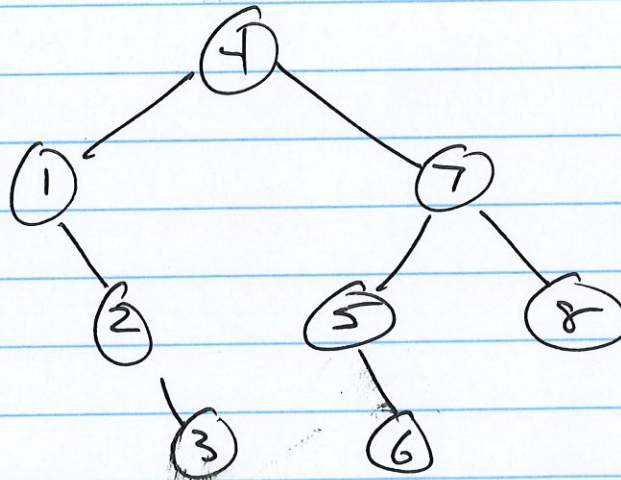
insert(3)



insert(5)



insert(6)

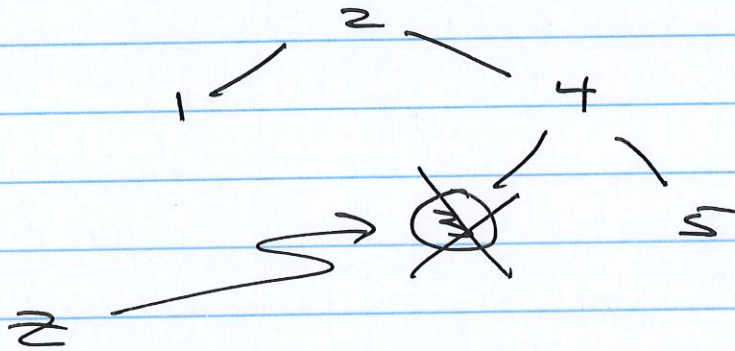


Runtime of insert
 $= \Theta(\text{height}(T))$

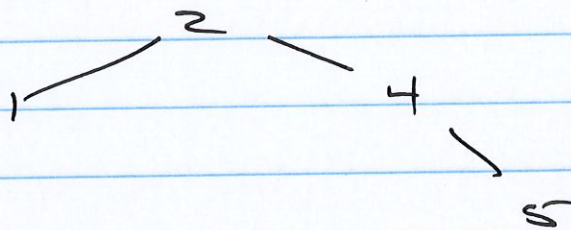
Deletion

delete $z \rightarrow$  from T .

Case 1 : z is a leaf



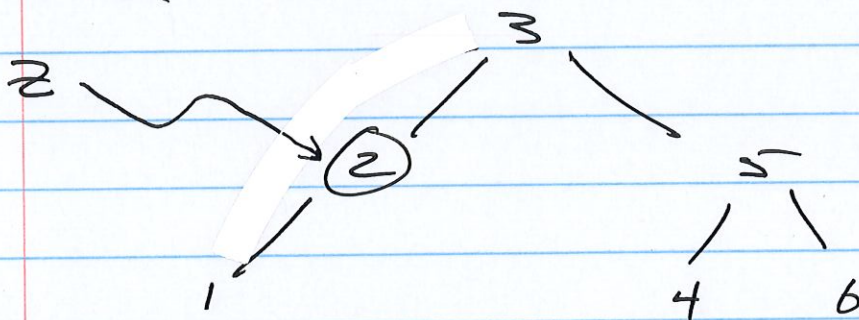
delete (3)



case 2 : z has 1 child

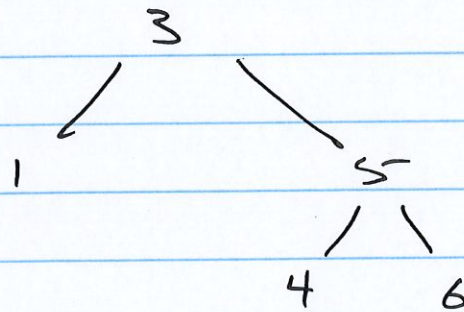
- (2.1) z has right but no left
 (2.2) z has left but no right

ex: case (2.2)



splice out z from 'linked list'

delete(z).



ex-ercise : draw subcase (2.1)

case 3: z has 2 children

splice out z 's successor
 y (which has no left child)

over-write z 's data with
that of y .