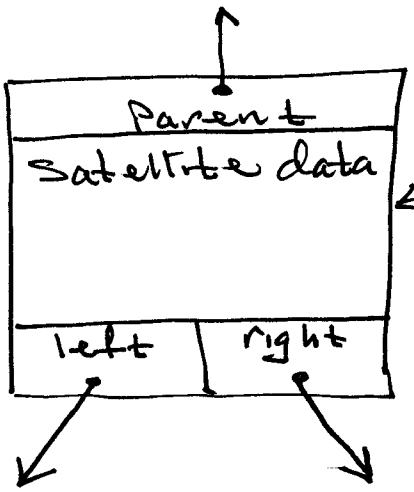
CSE 101 2-9-21

Part: 1 (absolutely last) day.

Binary Search Trees (BST)

Node Object



Dictionary: (key, val)

Simplify: key only.

key will be an integer in this discussion. In Pas, it is a string.

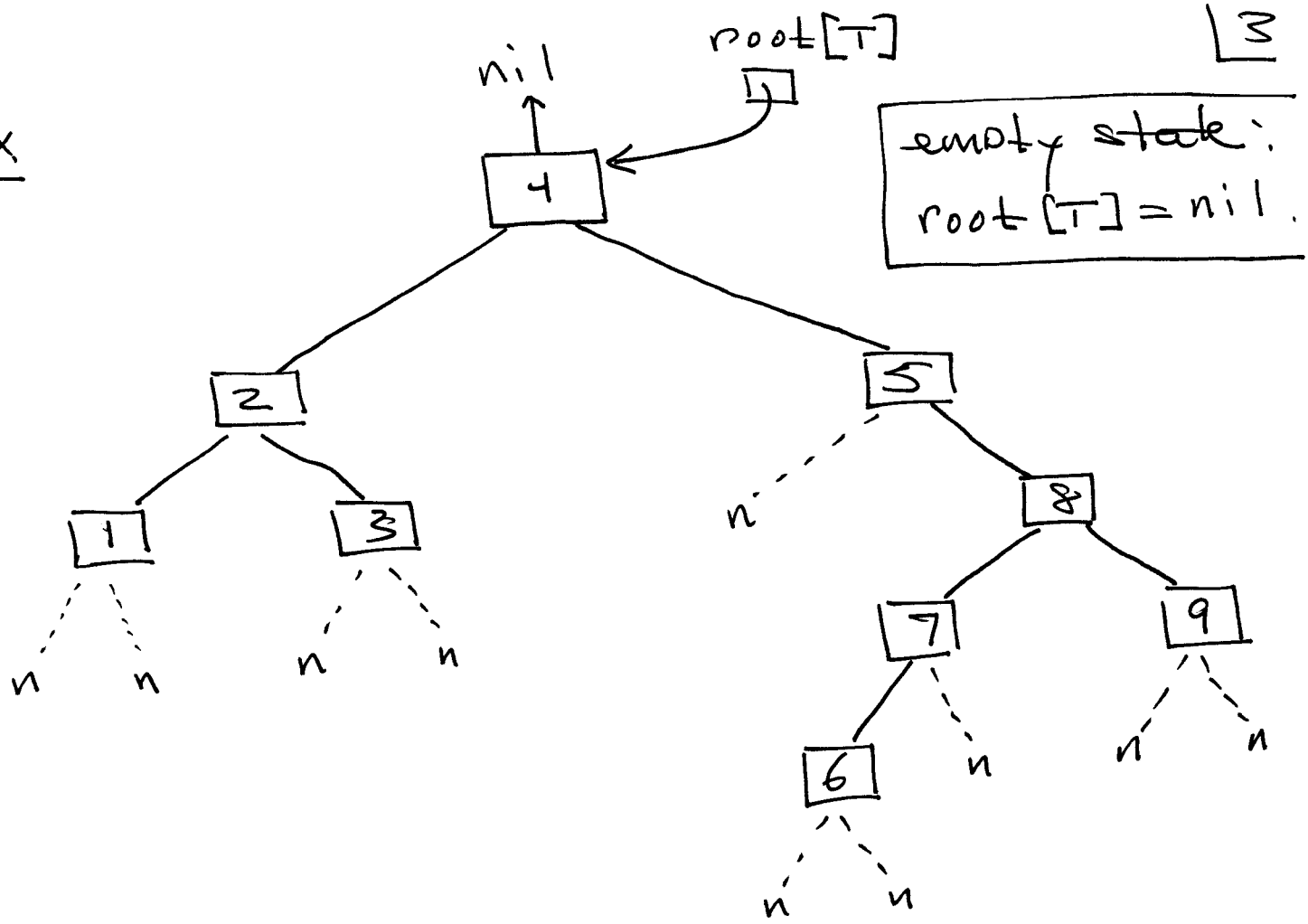
BST Properties

Let x, y be nodes in a BST. Then

(1) if y is in the left subtree of x (i.e. y is a descendant of left[x]) then: $key[y] \leq key[x]$

(2) if y is in the right subtree of x (i.e. a desc. of right[x]) then: $key[x] \leq key[y]$.

Ex



empty stack:
root[T] = nil.

Defn

- A leaf is a node with no children.
  {1, 3, 6, 9}
- An internal node is a non-leaf
  {4, 2, 5, 8, 7}

Recomendation: in Pas, create a dummy node called NIL as a field in DictionaryObj.

Tree Traversals

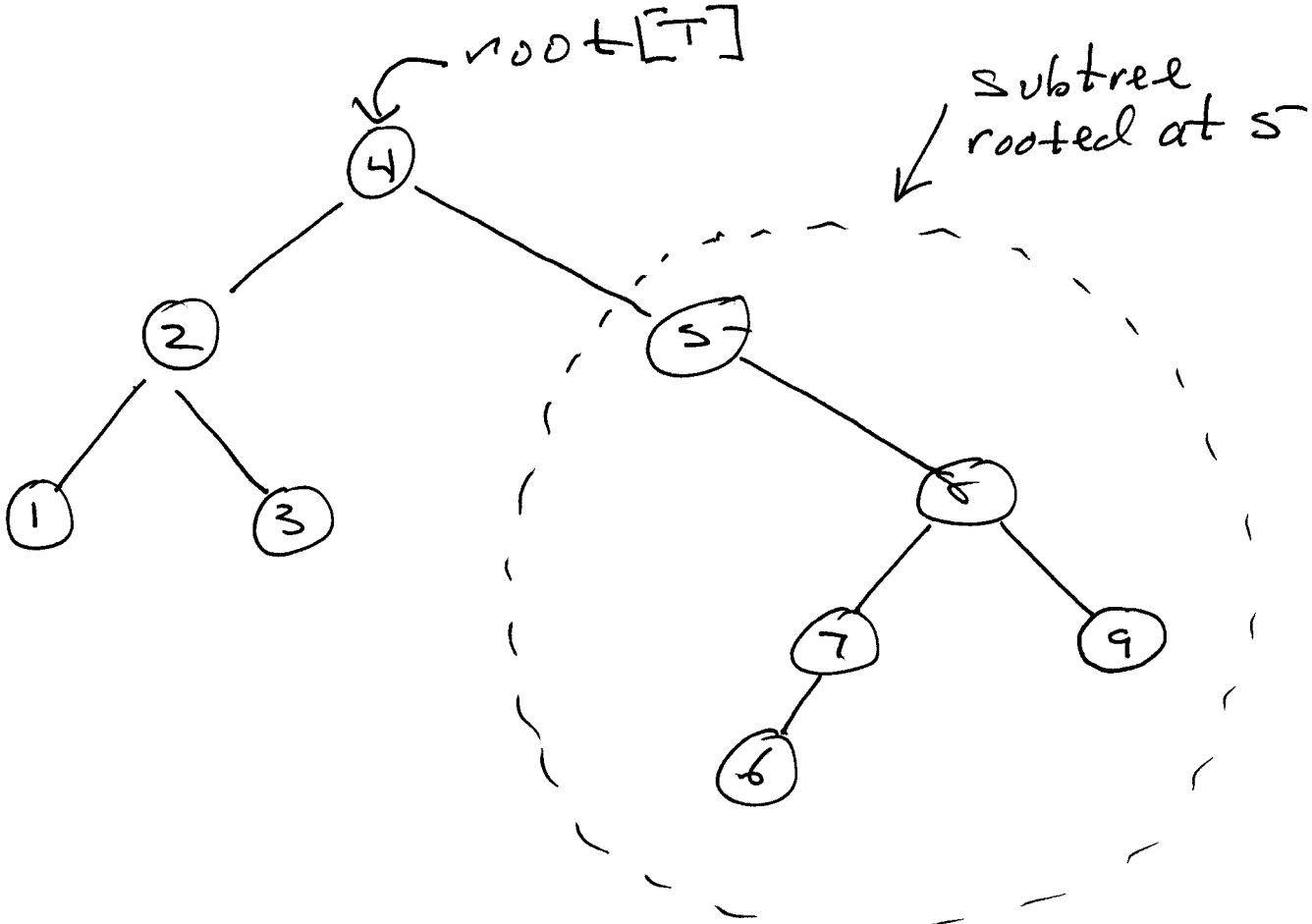
- InOrderTreeWalk(x)
- PreOrderTreeWalk(x)
- PostOrderTreeWalk(x)

Top level:

- InOrderTreeWalk(root[T])
  Output: 1 2 3 4 5 6 7 8 9
- PreOrderTreeWalk(root[T])
  Output: 4 2 1 3 5 8 7 6 9
- PostOrderTreeWalk(root[T])
  Output: 1 3 2 6 7 9 8 5 4

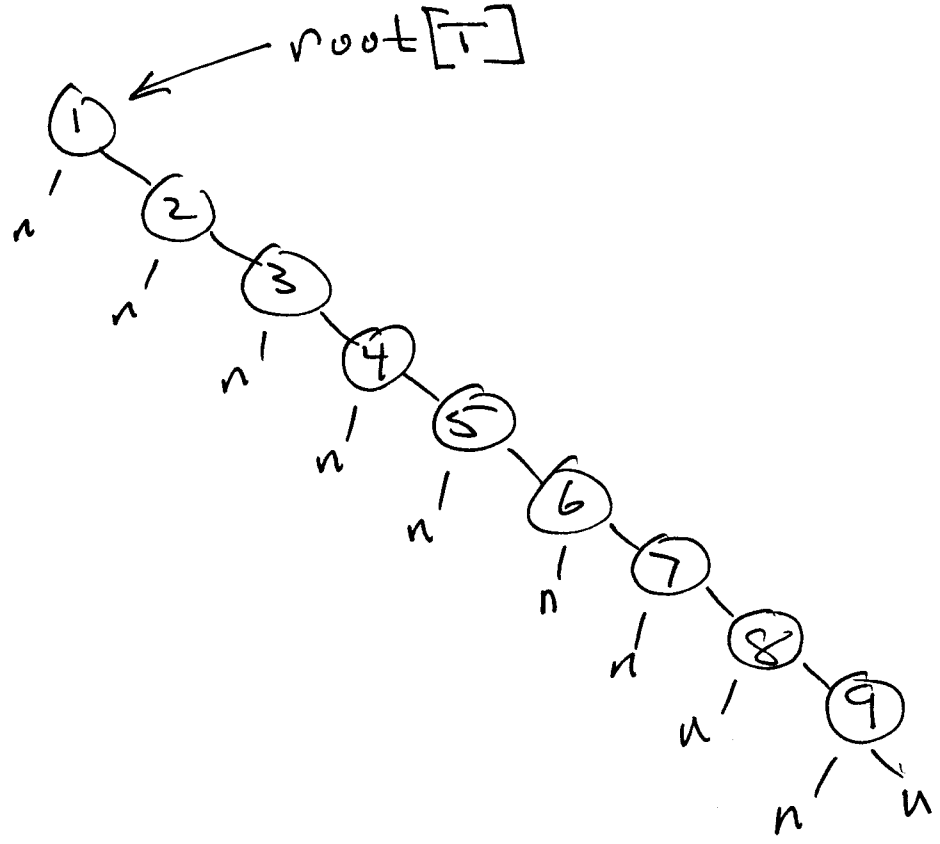
Queries

- TreeSearch(x, k)
- TreeMinimum(x)
- TreeMaximum(x)



Toplevel : TreeSearch(root[T], k)

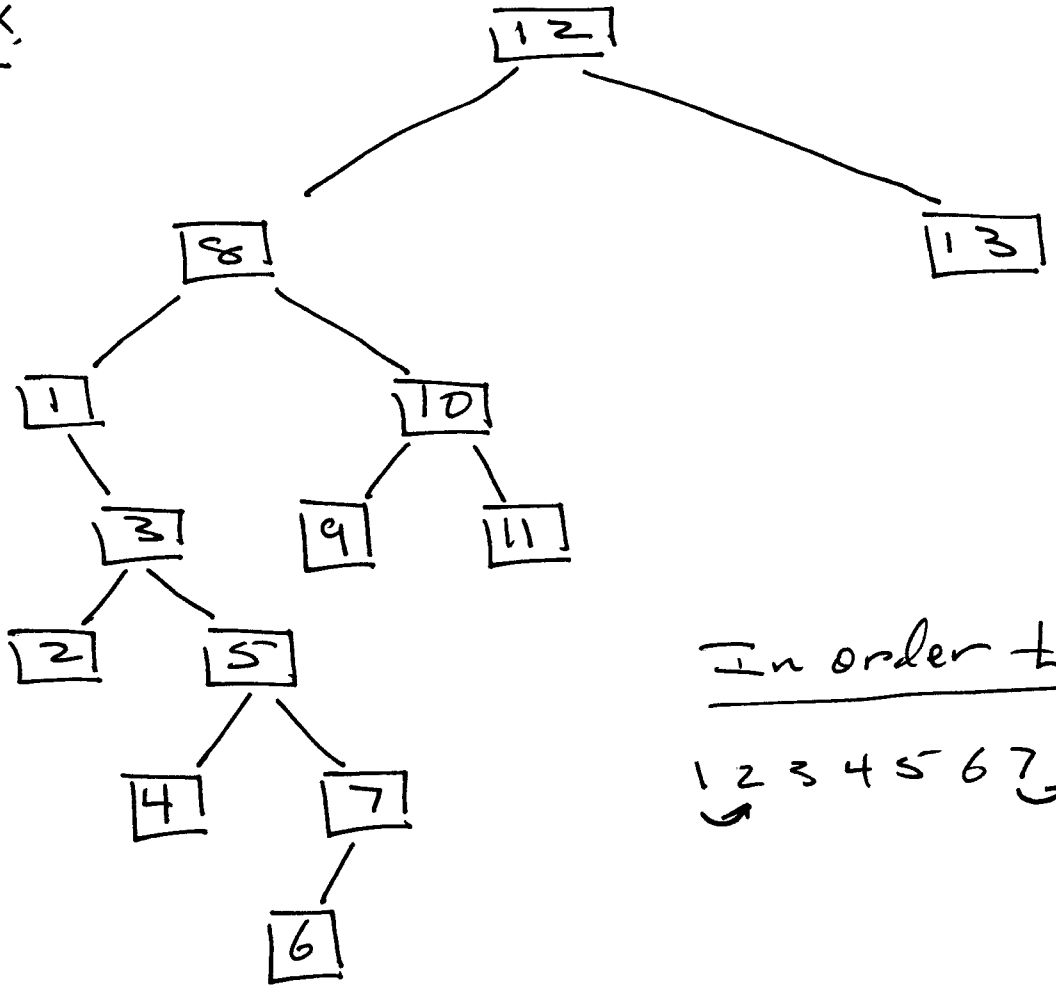
unbalanced tree



Def:

The successor of a node x is the next node after x to be printed/processed in an in-order tree walk.

Ex.



In order tree walk:

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)

Two cases:

case 1: x has a right child

case 2: x has no right child.

Past Picture