

(I moved and took a while)  
to get wife

\* reason for lateness

Kwame Kwah  
Tree Lab  
CMSC 204

### Slide 1

1) (a) Inorder traversal of tree:

In Inorder the order of traversal is first traverse left subtree then after travers Root and then traverse right subtree Right.

Inorder (Left, Root, Right): 16, 34, 35, 38, 39, 41, 44, 45, 55, 63, 64, 65, 72

(b) In preorder the order of traversal is first we traverse Root then after traverse left subtree and then traverse Right subtree.

Preorder (Root, Left, Right): 45, 38, 34, 16, 35, 41, 39, 44, 65, 63, 55, 64, 72

(c) In postorder the order of traversal is first traverse left subtree then after traverse Right subtree and then traverse Root.

Postorder (Left, Right, Root): ~~16, 35, 34, 39, 44, 41, 38, 55, 64, 63, 72, 65,~~  
~~45, 16, 35, 34, 39, 44, 41, 38, 55, 64, 63, 72, 65,~~  
~~45~~  
(45)

(d) The Height of the tree is 4

Nodes at level 2 is 38, 65

### Slide 2

(a) InOrder Traversal is:  $9 - 20 \% 9 / 42 * 23 - 64 * 6 + 12$

(b) Post Order Traversal is:  $9 \ 20 \ 9 \ \% \ - \ 42 / 23 \ 64 \ 6 \ * \ - \ 12 \ + \ *$

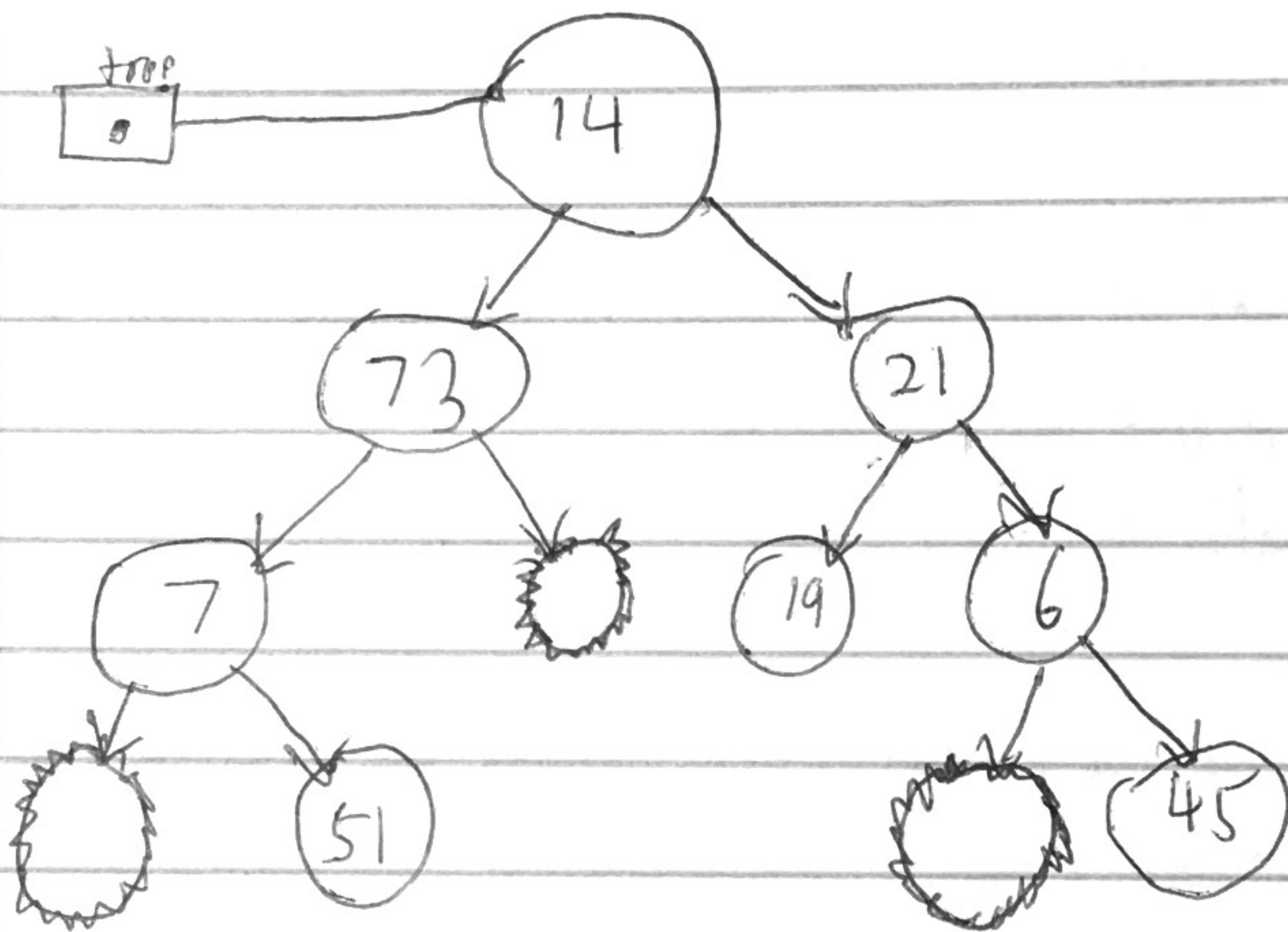
(c) If we use Integer Division:  $(9 - (2)) / 42 = 7 / 42 = 0$  [something] = 0, so it evaluates to 0

(d) if we use Floating Division:  $7 / 42 * 23 - 64 * 6 + 12 \Rightarrow 7 / 42 * (-349) \Rightarrow -58.167$



### Slide 3

- a) If the tree is not complete, a negative value (that is  $-1$ ) can be stored in the dummy value; or if the elements are objects, we can use a null value.



[0]	14
[1]	73
[2]	21
[3]	7
[4]	-
[5]	19
[6]	6
[7]	-
[8]	51
[9]	-1
[10]	45
[11]	
[12]	
[13]	
[14]	
[15]	

This figure is the Binary Tree stored in array with dummy values.



Slide 4

Array:

35	20	71	40	52	63	null	17	25	null	7	null	45
----	----	----	----	----	----	------	----	----	------	---	------	----

Tree: →

