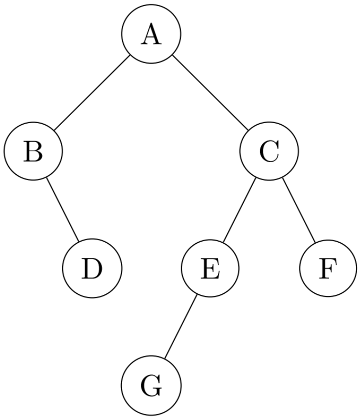
- This lab will cover Binary Trees.
- It is assumed that you have reviewed chapter 7 & 8 of the textbook. You may want to refer to the text and your lecture notes during the lab as you solve the problems.
- When approaching the problems, think before you code. Doing so is good practice and can help you lay out possible solutions.
- Think of any possible test cases that can potentially cause your solution to fail!
- If you finish early, you may leave early after showing the TA your work. Or you may stay and help other students. If you don't finish by the end of the lab, we recommend you complete it on your own time. Ideally you should not spend more time than suggested for each problem.
- Your TAs are available to answer questions in lab, during office hours, and on Edstem.

Vitamins

1. Given the following binary tree, complete the following (5 minutes):



a. Give the preorder, inorder, and postorder traversals of the tree:

Preorder:	A B D C E G F
Inorder:	B D A G E C F
Postorder:	D B G E F C A

b. Give the level order traversal (Breadth-First) of the tree:
A B C D E F G

c. What is the height of the tree?
3 (root is 0th level)

d. What are the depths of nodes E, B, and G?
E - 2 (there are 2 edges from root A to node E)
B - 1 (there are 1 edges from root A to node E)
G - 3 (there are 3 edges from root A to node E)

2. Draw the binary tree given the following traversals (10 minutes):

preorder : 11, 6, 4, 5, 8, 10, 19, 17, 43, 31, 49 CLR

inorder : 4, 5, 6, 8, 10, 11, 17, 19, 31, 43, 49 LCR

Draw the tree properly without guess and check

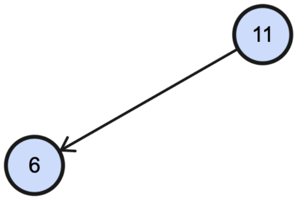
Identify the root : 11 (first in the preorder, last in the postorder)

- separate left and right in the inorder
- traverse preorder left to right and keep separating left and right

preorder : 11, 6, 4, 5, 8, 10, 19, 17, 43, 31, 49 (go from left to right)

inorder : 4, 5, 6, 8, 10, 11, 17, 19, 31, 43, 49
left right

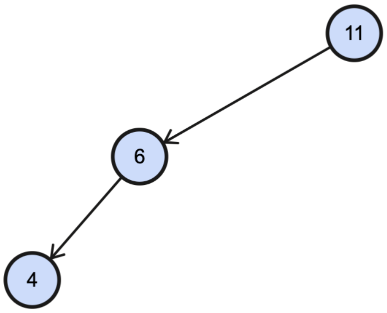
place the 11 as the root, the next number is 6, which is to the left of 11



preorder : ~~11, 6,~~ 4, 5, 8, 10, 19, 17, 43, 31, 49 (go from left to right)

inorder : 4, 5, ~~6~~, 8, 10, ~~11~~, 17, 19, 31, 43, 49
left right

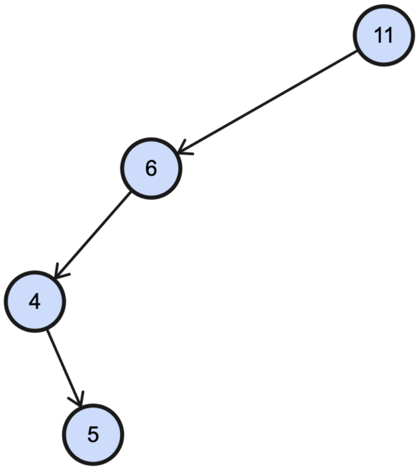
the next number is 4, which is to the left of 6



preorder : ~~11, 6, 4,~~ 5, 8, 10, 19, 17, 43, 31, 49 (go from left to right)

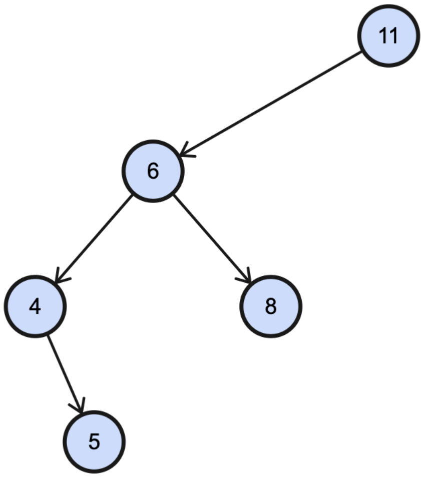
inorder : ~~4~~, 5, ~~6~~, 8, 10, ~~11~~, 17, 19, 31, 43, 49
right

the next number is 5, which is to the right of 4



preorder : ~~11, 6, 4, 5~~, 8, 10, 19, 17, 43, 31, 49 (go from left to right)

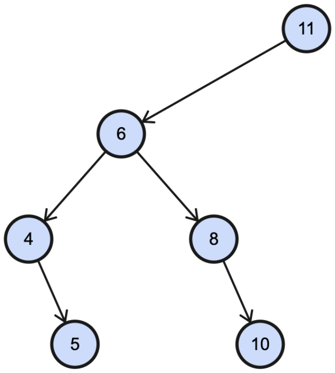
inorder : ~~4, 5, 6,~~ 8, 10, ~~11~~, 17, 19, 31, 43, 49 the next number is 8, which is to the right of 6



preorder : ~~11, 6, 4, 5, 8,~~ 10, 19, 17, 43, 31, 49 (go from left to right)

inorder : ~~4, 5, 6, 8,~~ 10~~, 11~~, 17, 19, 31, 43, 49

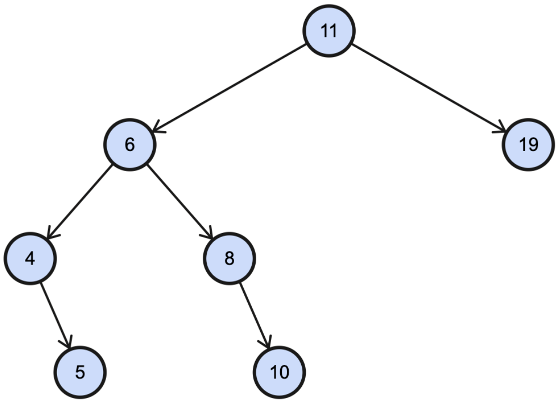
the next number is 10, which is to the right of 8



preorder : ~~11, 6, 4, 5, 8, 10~~, 19, 17, 43, 31, 49 (go from left to right)

inorder : ~~4, 5, 6, 8, 10, 11~~, 17, 19, 31, 43, 49

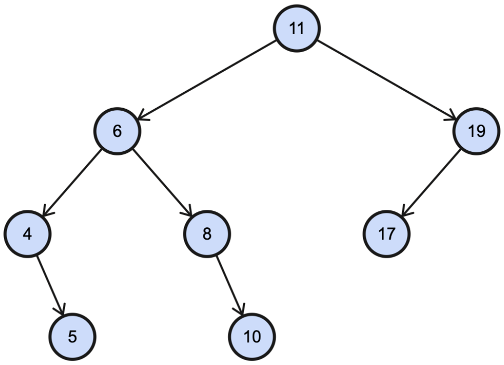
the next number is 11, which is to the right of 19



preorder : ~~11, 6, 4, 5, 8, 10, 19~~, 17, 43, 31, 49 (go from left to right)

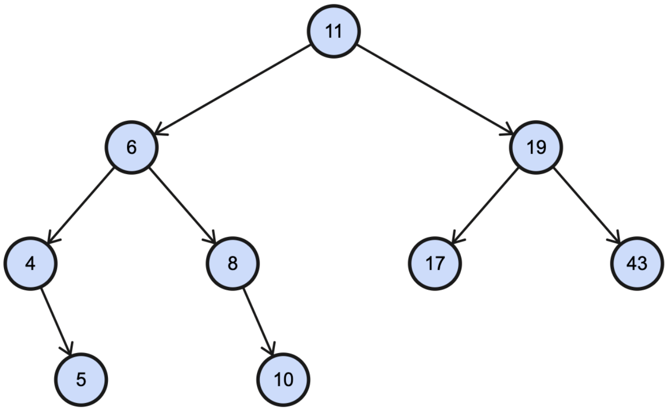
inorder : ~~4, 5, 6, 8, 10, 11~~, 17, ~~19~~, 31, 43, 49
left right

the next number is 17, which is to the left of 19



preorder : ~~11, 6, 4, 5, 8, 10, 19, 17,~~ 43, 31, 49 (go from left to right)

inorder : ~~4, 5, 6, 8, 10, 11, 17, 19~~, 31, 43, 49 the next number is 43, which is to the right of 19

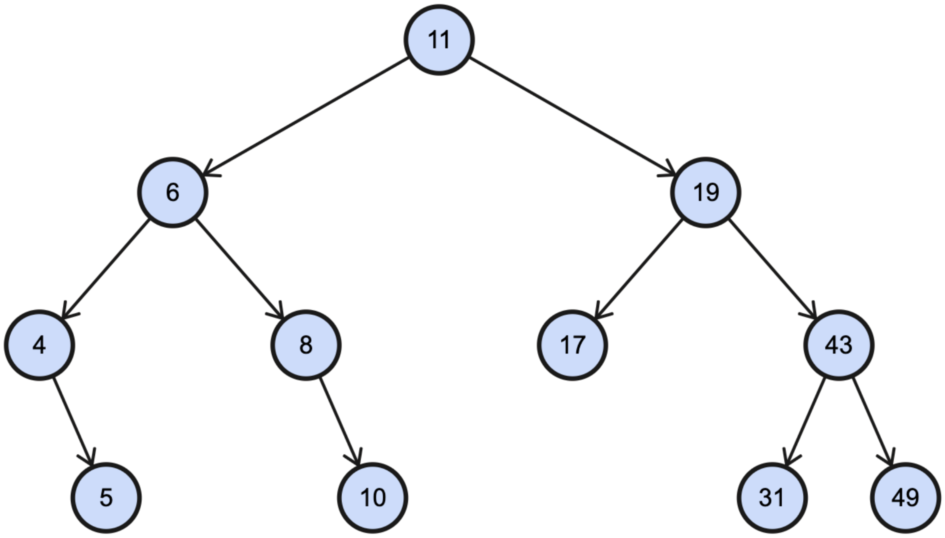


preorder : ~~11, 6, 4, 5, 8, 10, 19, 17, 43,~~ 31, 49 (go from left to right)

inorder : ~~4, 5, 6, 8, 10, 11, 17, 19~~, 31, ~~43,~~ 49

left right

31 is next, left of 43, 49 is next, right of 43, final result:



Is it possible to draw a unique binary tree given only its preorder and postorder? If not, draw two trees with the same preorder and postorder traversal.

No, the inorder allows you to distinguish between the left and right sections from the current node.

The preorder is CURR LEFT RIGHT, whereas postorder is LEFT RIGHT CURR. You can't distinguish between the left and right side.

If given:

preorder: 1, 2, 3
postorder: 3, 2, 1

you can draw a tree like so

```
        1
    2
3
```

or like so

```
1
    2
        3
```