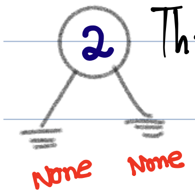
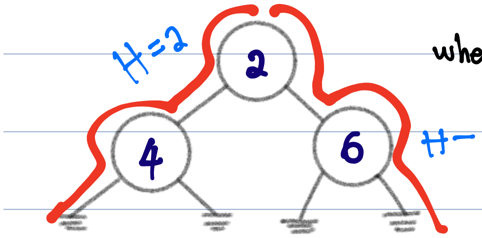
max height: the number of nodes from root node to the leaf node in a longest path.



This Tree's height is one, because there is only one to the leaf node



when there is path, the longest Path is the max height.

Note: It can be seen that when the root becomes none, we stop counting

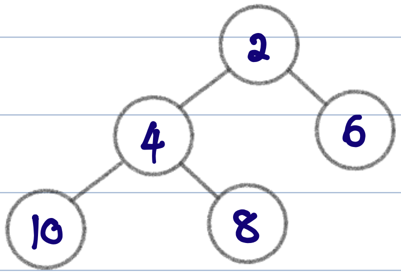
Algorithm:

① Traverse whole tree, both to the left and right of every node;

② add 1 to the height whenever you encounter a node.

③ stop moving further in a node when the root of that node becomes None.

```
def maxHeight(root):
    if root == None:
        return 0
    else:
        leftHeight = maxHeight(root.left)
        rightHeight = maxHeight(root.right)
        return max(leftHeight, rightHeight) + 1
```



```
maxHeight (2)
    maxHeight (4)
            4.left = 10
            LH = maxHeight (10)
                                  10.left
                LH = maxHeight (root.left) = maxHeight (None)
                                  10.right
                RH = maxHeight (root.right) = maxHeight (None)
            4.right = 8                         return max(0,0)+1
                                                 LH = 1
            RH = maxHeight (8)
                LH = maxHeight (8. left)  = maxHeight(None)
                RH = maxHeight (8.right) = maxHeight (None)
                return max(0,0)+1
                RH = 1
        return max(1,1) +1=2
  return max(2,1)+1 =3
```