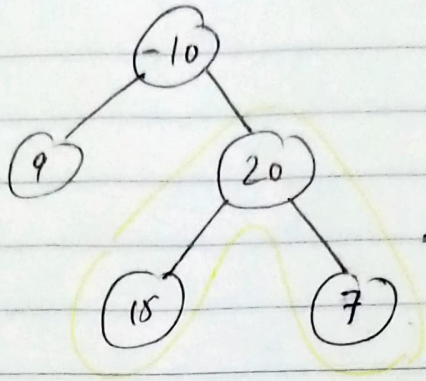
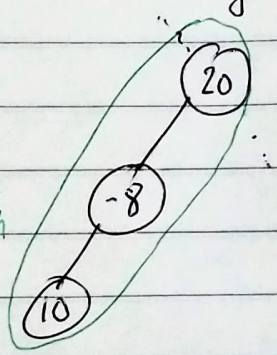
124: Binary Tree Maximum Path Sum



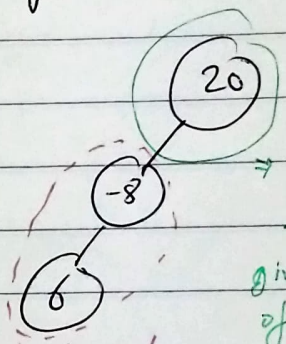
→ 20+15+7 = 42
maximum sum in continous path.

→ Nodes can have -ve numbers

→ If a path contain -ve nodes and if they don't add any value then they can be ignored.



Here -8 can be part of the path as total sum would be 22



→ Here choosing -8 and 6 gives total sum of 18.

can be ignored

→ **Four cases**

→ (1) Path involves only left ~~child~~ side
(2) path involves only right side.
(3) path includes both left and right
(4) path only involves root node (i.e both left and right side yield -ve sums)

(Post-order traversal)

```
def func(~~nod~~ root)
    max_sum = 0

    def gain_from_subtree(root):
        if not node: return 0
        left_gain = max(gain_from_subtree(root.left), 0)
        right_gain = max(gain_from_subtree(root.right), 0)
        max_sum = max(max_sum, left_gain + right_gain
                                + root.val)
        return max(left_gain + root.val, right_gain + root.val)
```

Code is very simple