Burn the binary tree starting from the target node

Given a binary tree and target node. By giving the fire to the target node and fire starts to spread in a complete tree.

Find the time to burn entire tree.

```
import sys


class TreeNode:
    def __init__(self, val):
        self.data = val
        self.left = None
        self.right = None

def burningTree(root, target,maxTime):
    if root is None:
        return -1
    if root.data == target:
        burnKDown(root, None, 0,maxTime)
        return 1
    leftTime = burningTree(root.left, target,maxTime)
    if leftTime != -1:
        burnKDown(root, root.left, leftTime,maxTime)
        return leftTime + 1
    rightTime = burningTree(root.right, target,maxTime)
    if rightTime != -1:
        burnKDown(root, root.right, rightTime,maxTime)
        return rightTime + 1
    return -1


def burnKDown(root, blockNode, time,maxTime):
    if root is None or root is blockNode:
        return
    maxTime[0] = max(maxTime[0], time)
    burnKDown(root.left, blockNode, time + 1,maxTime)
    burnKDown(root.right, blockNode, time + 1,maxTime)
```

```
maxTime = [0]

root = TreeNode(10)
root.left = TreeNode(12)
root.right = TreeNode(13)

root.right.left = TreeNode(14)
root.right.right = TreeNode(15)

root.right.left.left = TreeNode(21)
root.right.left.right = TreeNode(22)
root.right.right.left = TreeNode(23)
root.right.right.right = TreeNode(24)

res = []
print(burningTree(root, 14,maxTime))
# print(res)
```