Print 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. The task is to print the sequence of the burning nodes of a binary tree.
Rules for burning the nodes :

- Fire will spread constantly to the connected nodes only.
- Every node takes the same time to burn.
- A node burns only once.

```
import sys


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




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

```
def burnKDown(root, blockNode, time,nodesBurnt):
    if root is None or root is blockNode:
        return
    if time in nodesBurnt:
        nodesBurnt[time].append(root.data)
    else:
        nodesBurnt[time] = [root.data]
    burnKDown(root.left, blockNode, time + 1,nodesBurnt)
    burnKDown(root.right, blockNode, time + 1,nodesBurnt)


nodesBurnt = {}

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)

print(burningTree(root, 14,nodesBurnt))
print(nodesBurnt)
```