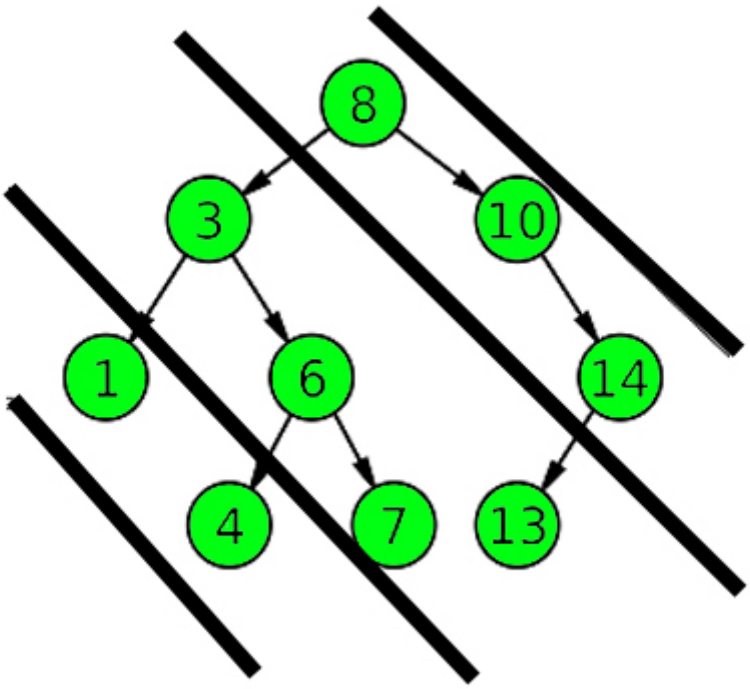
Diagonal Traversal of Binary Tree

Given a Binary Tree, print the **diagonal traversal** of the binary tree.

Consider lines of slope -1 passing between nodes. Given a Binary Tree, print all diagonal elements in a binary tree belonging to same line.

Input :
8
/
3 10
/
1 6 14
/ \ /
4 7 13
Output : 8 10 14 3 6 7 13 1 4
Explanation:



Diagonal Traversal of binary tree :
8 10 14 3 6 7 13 1 4

```
def diagonal(root):
    res = {}
    helper(root,0,res)
    i = 0
    ans = []
```

```
    while True:
        if i in res:
            ans = ans+res[i]
            i = i+1
        else:
            break
    return ans

def helper(root,level,res):
    if root is None:
        return
    if level not in res:
        res[level] = [root.data]
    else:
        res[level] =  res[level]+[root.data]

    helper(root.left,level+1,res)
    helper(root.right,level,res)
```

Hint:

If you move to left increment the count, and for a particular count print the answer one after the other.

```
import collections
import sys

def diagonal(root):
    #:param root: root of the given tree.
    #return: print out the diagonal traversal,  no need to print new line
    #code here
    ans = collections.defaultdict(list)
    helperDiagonal(root,ans,0)
    res = []
    idx = 0
    while idx in ans:
        res.extend(ans[idx])
        idx-=1
    return res

def helperDiagonal(root,ans,level):
    if root is None:
        return
    if level in ans:
        ans[level].append(root.data)
```

```
    else:
        ans[level] = [root.data]
    helperDiagonal(root.left,ans,level-1)
    helperDiagonal(root.right,ans,level)
```