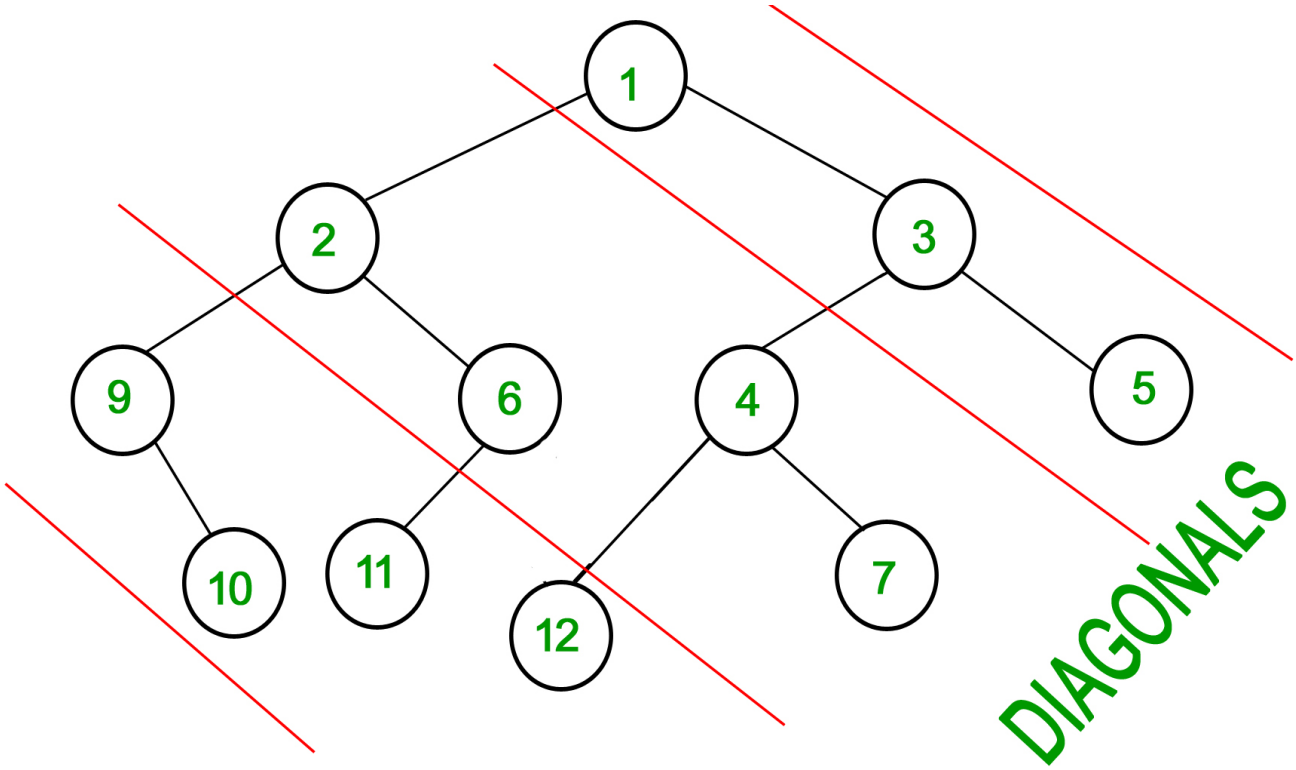
Diagonal Sum of a Binary Tree

Consider lines of slope -1 passing between nodes (dotted lines in below diagram). The diagonal sum in a binary tree is the sum of all node's data lying between these lines. Given a Binary Tree, print all diagonal sums.
For the following input tree, the output should be 9, 19, 42.
9 is sum of 1, 3 and 5.
19 is sum of 2, 6, 4 and 7.
42 is sum of 9, 10, 11 and 12.




```
# Program to find diagonal sum in a Binary Tree

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

# Function to compute height and
# root - root of the binary tree
# vd - vertical distance diagonally
# diagonalSum - map to store Diagonal
# Sum(Passed by Reference)
def diagonalSumUtil(root, vd, diagonalSum) :
```

```
    if(not root):
        return

    if vd not in diagonalSum:
        diagonalSum[vd] = 0
    diagonalSum[vd] += root.data

    # increase the vertical distance
    # if left child
    diagonalSumUtil(root.left, vd + 1,
                        diagonalSum)

    # vertical distance remains same
    # for right child
    diagonalSumUtil(root.right, vd,
                    diagonalSum)

# Function to calculate diagonal
# sum of given binary tree
def diagonalSum(root) :

    # create a map to store Diagonal Sum
    diagonalSum = dict()

    diagonalSumUtil(root, 0, diagonalSum)

    print("Diagonal sum in a binary tree is - ",
                                    end = "")

    for it in diagonalSum:
        print(diagonalSum[it], end = " ")

# Driver Code
if __name__ == '__main__':
    root = newNode(1)
    root.left = newNode(2)
    root.right = newNode(3)
    root.left.left = newNode(9)
    root.left.right = newNode(6)
    root.right.left = newNode(4)
    root.right.right = newNode(5)
    root.right.left.right = newNode(7)
    root.right.left.left = newNode(12)
```

```
    root.left.right.left = newNode(11)
    root.left.left.right = newNode(10)

    diagonalSum(root)

# This code is contributed
# by SHUBHAMSINGH10
```