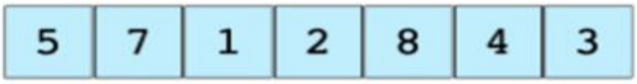
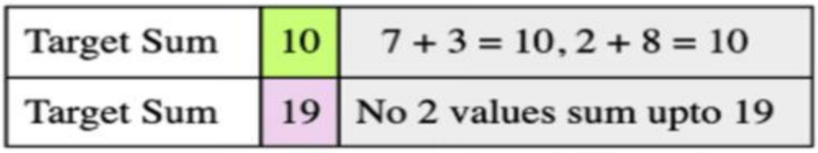
Google algorithms interview question and answer

1. Given an array of integers and a value, determine if there are any two integers in the array whose sum is equal to the given value. Return true if the sum exists and return false if it does not.



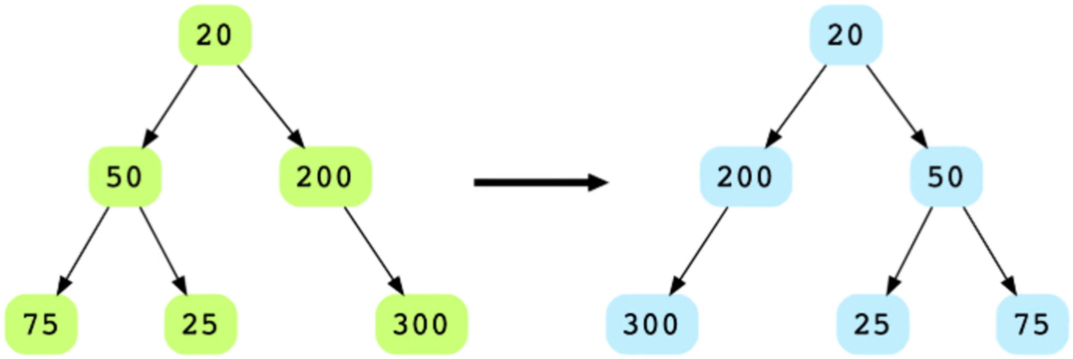
5	7	1	2	8	4	3



Target Sum	10	7 + 3 = 10, 2 + 8 = 10
Target Sum	19	No 2 values sum upto 19

```
</div>
<div>def find_sum_of_two(A, val):
found_values = set()
for a in A:
if val - a in found_values:
returnTrue
found_values.add(a)
returnFalse
v = [5,7,1,2,8,4,3]
test = [3,20,1,2,7]
for i inrange(len(test)):
output = find_sum_of_two(v, test[i])
print("find_sum_of_two(v, " + str(test[i]) + ") = " +
str(output))</div>
</div>
<div>
```

2. Given the root node of a binary tree, swap the ‘left’ and ‘right’ children for each node. The below example shows how the mirrored binary tree should look like.



```

</div>
<div>def mirror_tree(root):
if root == None:
return
# We will do a post-order traversal of the binary tree.
if root.left != None:
mirror_tree(root.left)
if root.right != None:
mirror_tree(root.right)
# Let's swap the left and right nodes at current level.
temp = root.left
root.left = root.right
root.right = temp
def level_order_traversal(root):
if root == None:
return
q = deque()
q.append(root)
while q:
temp = q.popleft()
print(str(temp.data), end = ",")
if temp.left != None:
q.append(temp.left)
if temp.right != None:
q.append(temp.right)
arr = [100,25,75,15,350,300,10,50,200,400,325,375]
root = create_BST(arr)
#root = create_random_BST(15)
print("nLevel Order Traversal:", end = "")
level_order_traversal(root)
mirror_tree(root)
print("nMirrored Level Order Traversal:", end = "")
level_order_traversal(root)</div>
```