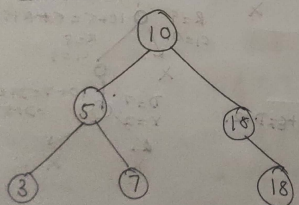
## Expt 4:- BFS & DFS for Real World Application Problems

**Problem Formulation** :- A graph with no. of nodes (BST) is given we have to find the sum of all the nodes in a given range specified by the user.

* **Path cost:-** Depends on no. of nodes traversed in a weighted or unweighted BST

* **Operators:-** The nodes and paths are the operators for the sum and traversal

* **Initial state:-**




(L = 7, H = 15).

* **Algorithm:-**

(1) BFS OR DFS:

(.) We traverse the tree using BFS & DFS.

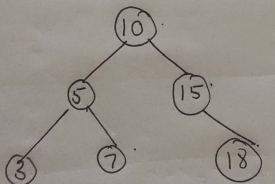
⇨ If node value falls outside the range [L, R] we know we only the right branch could have nodes with value insia [L, R]

⇨ For a recursive implement ation, the recursion will consume additional space in the functional call stack. In the worst case the tree is of chain shape, and we will reach all the way down to the leaf node.

⇨ Time Complexity:- O(N) O(N) where N is the number of nodes in the tree.

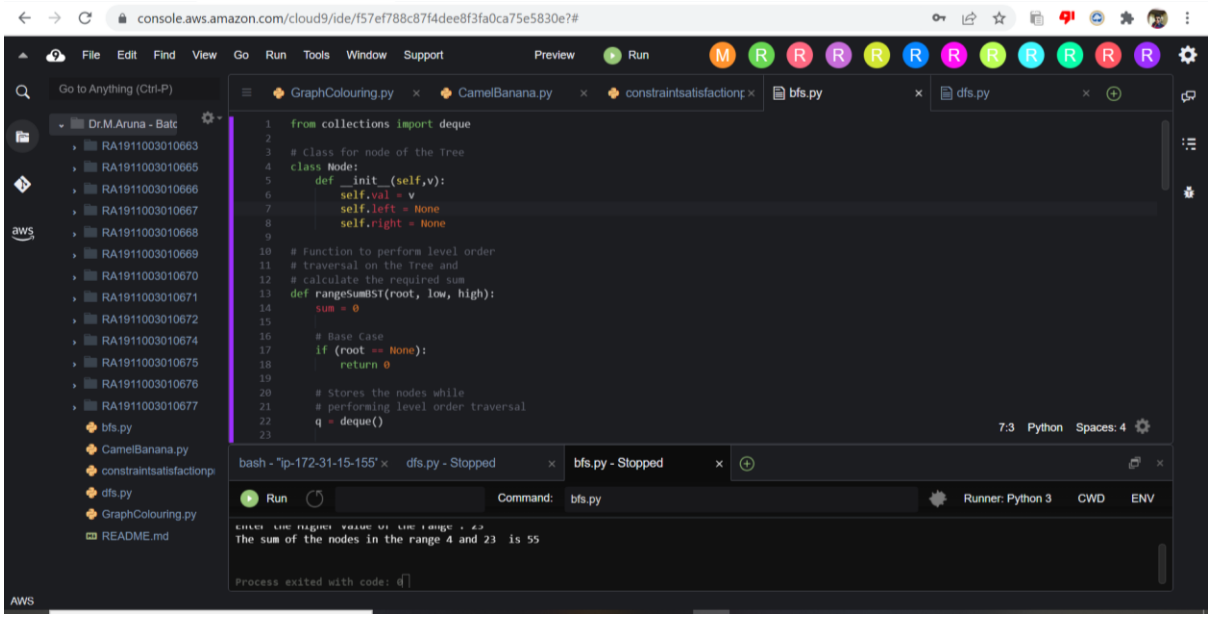
⇨ Space Complexity:- O(N) O(N)

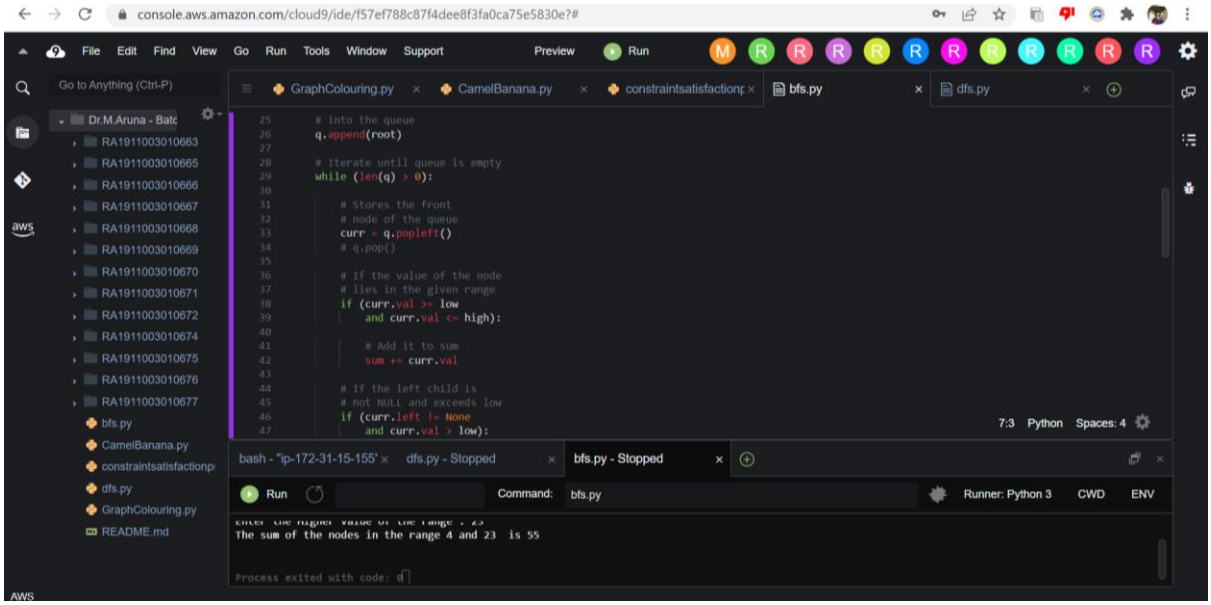
* Solution:-

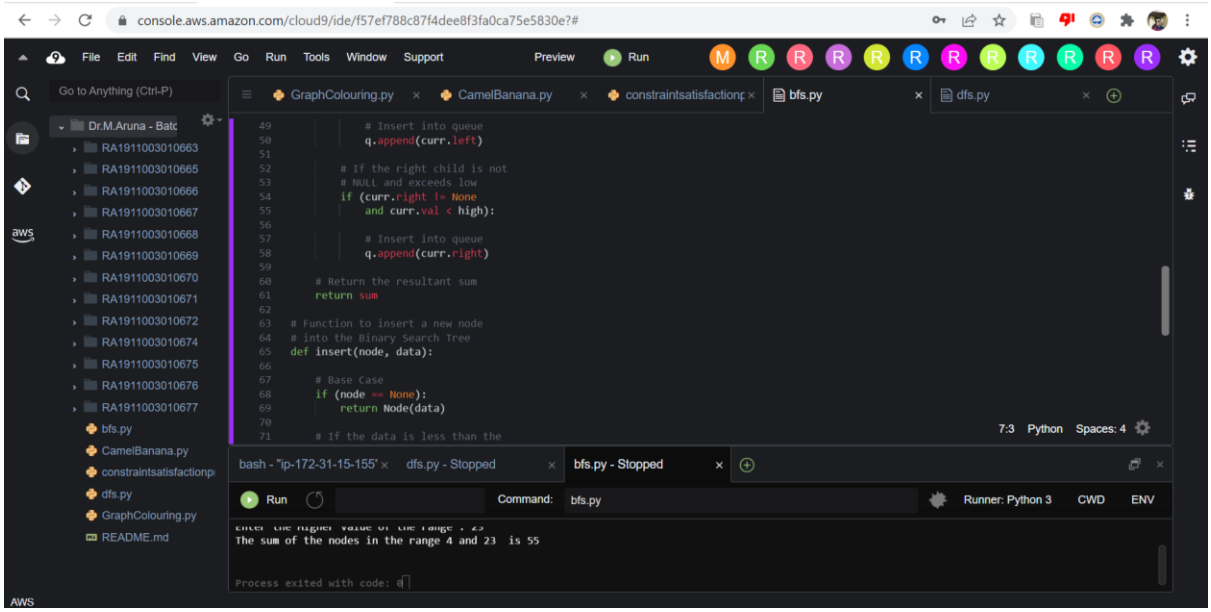


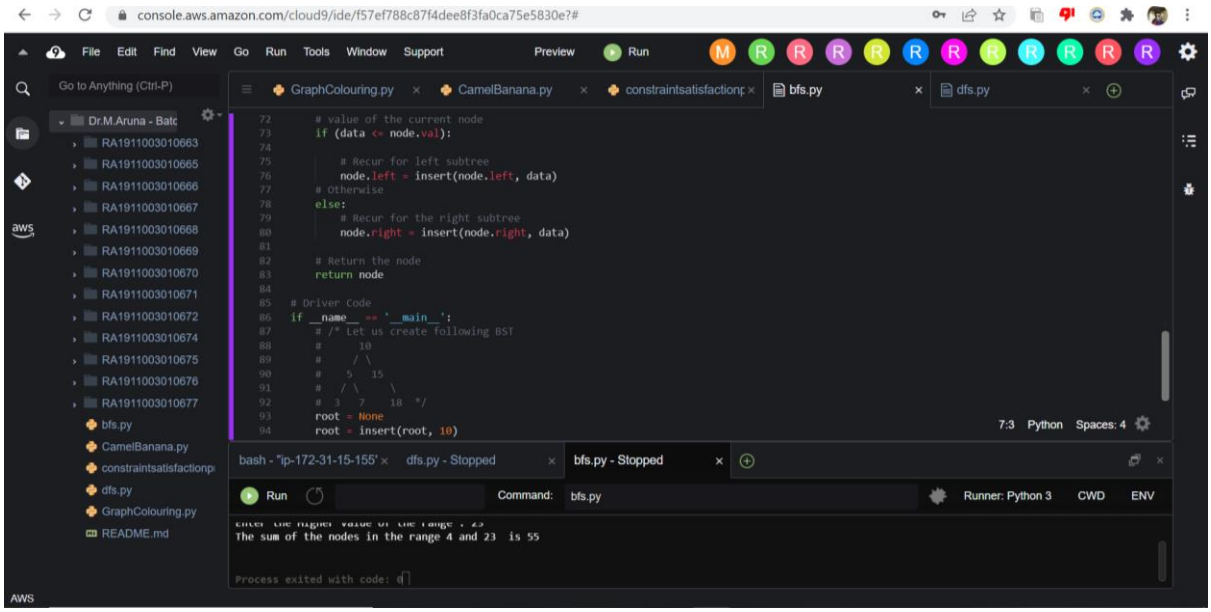

(10+15+7 = 32).

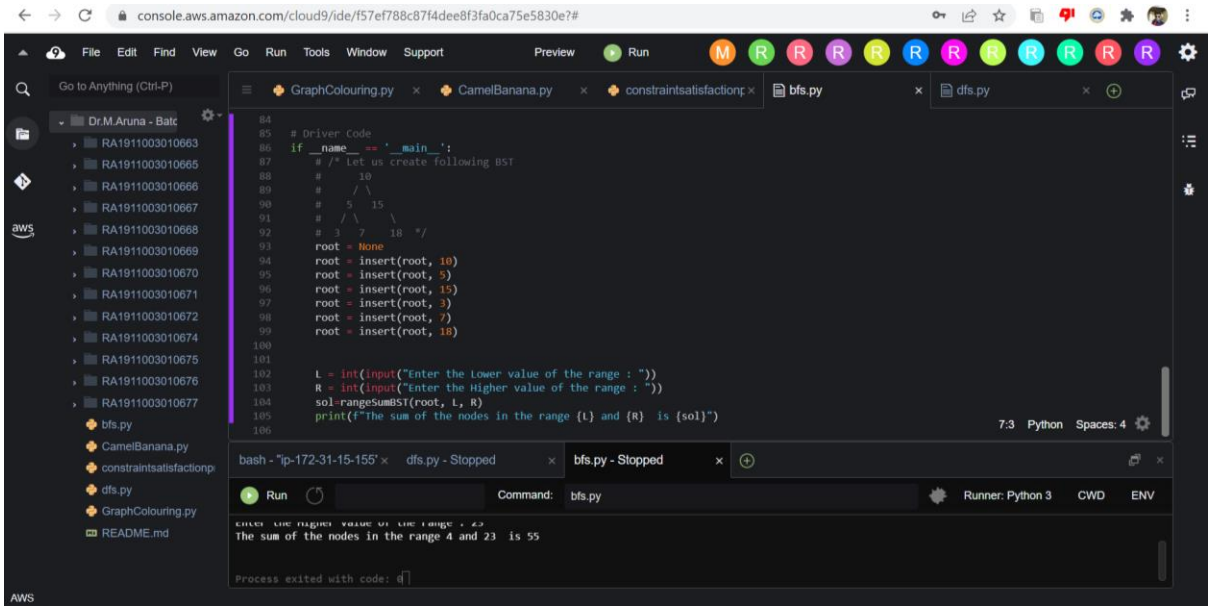
(Answer)

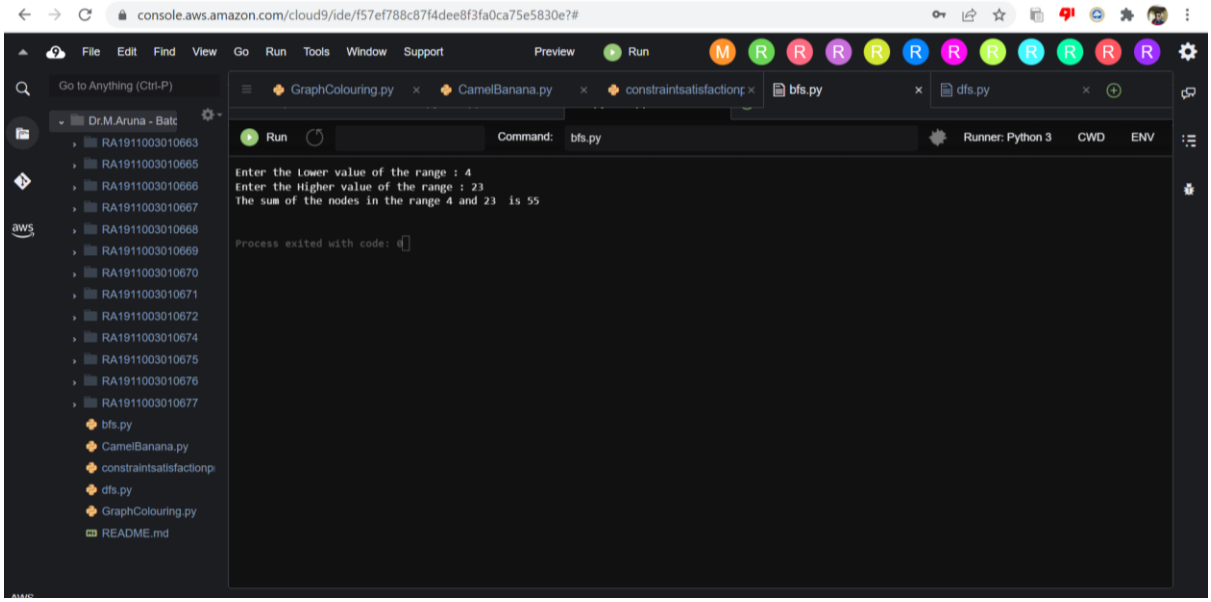

console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py CamelBanana.py constraintsatisfactionp bfs.py dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
from collections import deque
# Class for node of the Tree
class Node:
def __init__(self,v):
self.val = v
self.left = None
self.right = None
# Function to perform level order
# traversal on the Tree and
# calculate the required sum
def rangeSumBST(root, low, high):
sum = 0
# Base Case
if (root == None):
return 0
# Stores the nodes while
# performing level order traversal
q = deque()
7:3 Python Spaces: 4
bash - "ip-172-31-15-155" dfs.py - Stopped bfs.py - Stopped
Run Command: bfs.py Runner: Python 3 CWD ENV
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0
AWS

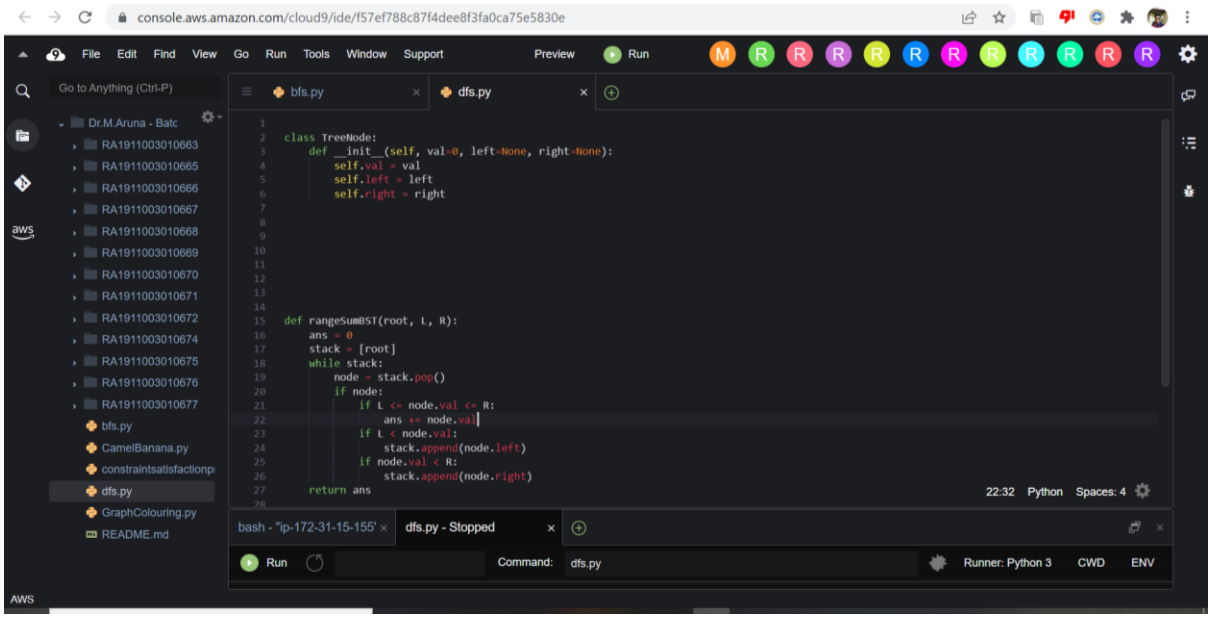

console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py CamelBanana.py constraintsatisfactionp bfs.py dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
# into the queue
q.append(root)
# Iterate until queue is empty
while (len(q) > 0):
# Stores the front
# node of the queue
curr = q.popleft()
# q.pop()
# If the value of the node
# lies in the given range
if (curr.val >= low
and curr.val <= high):
# Add it to sum
sum += curr.val
# If the left child is
# not NULL and exceeds low
if (curr.left != None
and curr.val > low):
7:3 Python Spaces: 4
bash - "ip-172-31-15-155" dfs.py - Stopped bfs.py - Stopped
Run Command: bfs.py Runner: Python 3 CWD ENV
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0
AWS

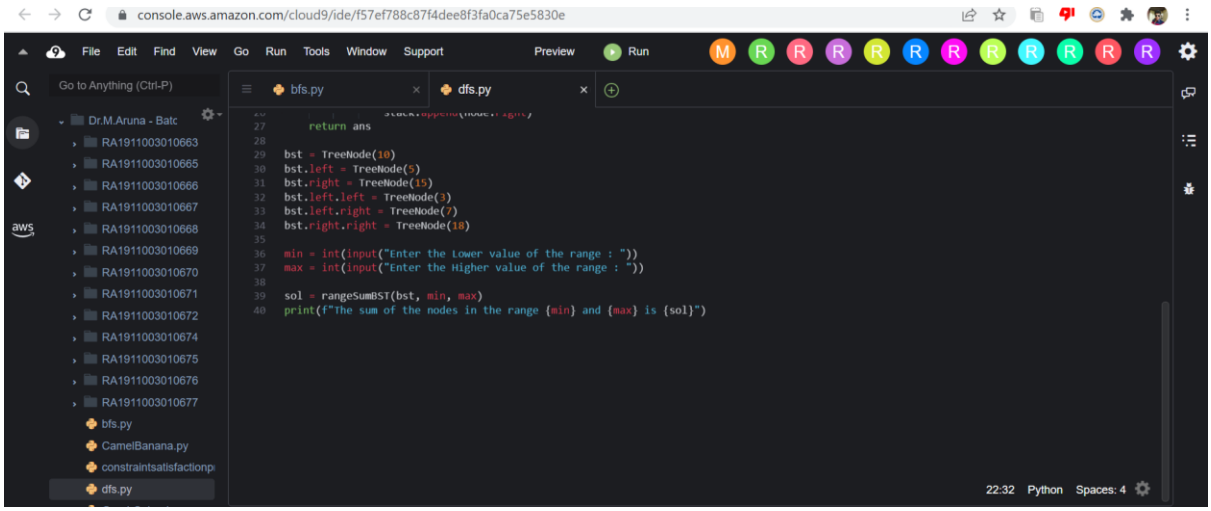

console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py
CamelBanana.py
constraintsatisfactionp
bfs.py
dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
# Insert into queue
q.append(curr.left)
# If the right child is not
# NULL and exceeds low
if (curr.right != None
and curr.val < high):
# Insert into queue
q.append(curr.right)
# Return the resultant sum
return sum
# Function to insert a new node
# into the Binary Search Tree
def insert(node, data):
# Base Case
if (node == None):
return Node(data)
# If the data is less than the
7:3 Python Spaces: 4
bash - "ip-172-31-15-155" dfs.py - Stopped bfs.py - Stopped
Run Command: bfs.py Runner: Python 3 CWD ENV
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0
AWS

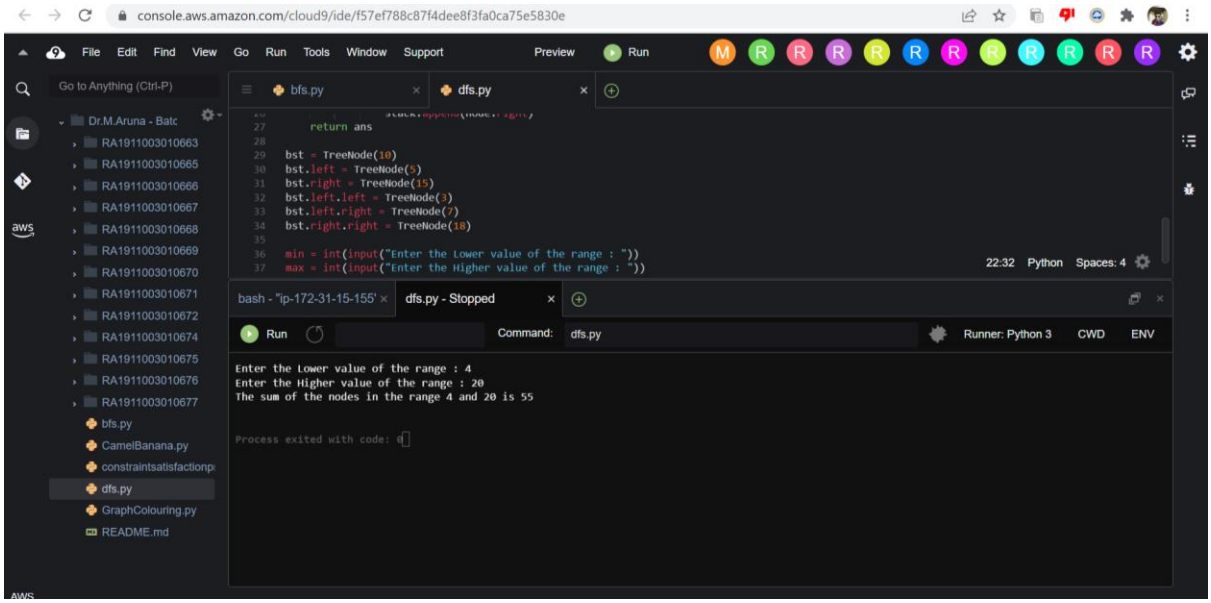

console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py
CamelBanana.py
constraintsatisfactionp
bfs.py
dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
# value of the current node
if (data <= node.val):
# Recur for left subtree
node.left = insert(node.left, data)
# Otherwise
else:
# Recur for the right subtree
node.right = insert(node.right, data)
# Return the node
return node
# Driver Code
if __name__ == '__main__':
# /* Let us create following BST
# 10
# / \
# 5 15
# / \ \
# 3 7 18 */
root = None
root = insert(root, 10)
7:3 Python Spaces: 4
bash - "ip-172-31-15-155" dfs.py - Stopped bfs.py - Stopped
Run Command: bfs.py Runner: Python 3 CWD ENV
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0
AWS


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py
CamelBanana.py
constraintsatisfaction
bfs.py
dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
# Driver Code
if __name__ == '__main__':
# /* Let us create following BST
# 10
# / \
# 5 15
# / \ \
# 3 7 18 */
root = None
root = insert(root, 10)
root = insert(root, 5)
root = insert(root, 15)
root = insert(root, 3)
root = insert(root, 7)
root = insert(root, 18)
L = int(input("Enter the Lower value of the range : "))
R = int(input("Enter the Higher value of the range : "))
sol=rangeSumBST(root, L, R)
print(f"The sum of the nodes in the range {L} and {R} is {sol}")
7:3 Python Spaces: 4
bash - "ip-172-31-15-155" x
dfs.py - Stopped
bfs.py - Stopped
Run
Command: bfs.py
Runner: Python 3 CWD ENV
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0
AWS


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
GraphColouring.py
CamelBanana.py
constraintsatisfaction
bfs.py
dfs.py
Dr.M.Aruna - Batc
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
bfs.py
CamelBanana.py
constraintsatisfactionp
dfs.py
GraphColouring.py
README.md
Run
Command: bfs.py
Runner: Python 3 CWD ENV
Enter the Lower value of the range : 4
Enter the Higher value of the range : 23
The sum of the nodes in the range 4 and 23 is 55
Process exited with code: 0


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
bfs.py
dfs.py
class TreeNode:
def __init__(self, val=0, left=None, right=None):
self.val = val
self.left = left
self.right = right
def rangeSumBST(root, L, R):
ans = 0
stack = [root]
while stack:
node = stack.pop()
if node:
if L <= node.val <= R:
ans += node.val
if L < node.val:
stack.append(node.left)
if node.val < R:
stack.append(node.right)
return ans
22:32 Python Spaces: 4
bash - "ip-172-31-15-155'
dfs.py - Stopped
Run
Command: dfs.py
Runner: Python 3
CWD
ENV
AWS


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
bfs.py
dfs.py
return ans
bst = TreeNode(10)
bst.left = TreeNode(5)
bst.right = TreeNode(15)
bst.left.left = TreeNode(3)
bst.left.right = TreeNode(7)
bst.right.right = TreeNode(18)
min = int(input("Enter the Lower value of the range : "))
max = int(input("Enter the Higher value of the range : "))
sol = rangeSumBST(bst, min, max)
print(f"The sum of the nodes in the range {min} and {max} is {sol}")
22:32 Python Spaces: 4




Result:-

The DFS and BFS on a real world problems has been successfully implemented.