DFS ALGORITHM IMPLEMENTATION USING STACK

Tree Structure:

tree = {

'A': ['B', 'C'],

'B': ['D', 'E'],

'C': ['F'],

'D': [],

'E': [],

'F': ['G'],

'G': []

}

DFS Stack Class:

class DFS\_Stack():

def \_\_init\_\_(self):

self.data = []

def push(self, x):

self.data.append(x)

def pop(self):

if len(self.data) > 0:

return self.data.pop()

else:

print("Your Data is Empty.")

DFS Function:

def dfs():

visit\_node = []

s = DFS\_Stack()

s.push("B")

while len(s.data) > 0:

a = s.pop()

print(f"Your node {a} is Remove.")

if a not in visit\_node:

visit\_node.append(a)

for c in reversed(tree[a]):

s.push(c)

return visit\_node

Output:

print(dfs())

Execution Result:

Your node B is Remove.

Your node D is Remove.

Your node E is Remove.

['B', 'D', 'E']