



Course Name: AIML Lab

Course Code: 21CSH-316

## **Experiment:1.2**

**Student Name:** UTKARSH JOSHI

**UID:** 21BCS9158

**Branch:** BE-CSE

**Section/Group:** 21BCS\_ST 802 “A”

**Semester:** 5<sup>th</sup>

**Date of Performance:** 17/08/23

**Subject Name:** AI&ML with Lab

**Subject Code:** 21CSH-316

**1. Aim:** Implement the DFS algorithm and analyze its performance and characteristics.

**2. Software used:** Virtual Studio CODE

**3. Pseudo code/Algorithms/Flowchart/Steps:**

```
Mytree = {
'2' : ['4', '6'],
'4' : ['8', '12', '16'],
'6' : ['36'],
'8' : ['18', '20'],
'12' : ['22'],
'16' : ['4'],
'36' : ['6'],
'18' : ['8'],
'20' : ['8'],
'22' : ['12'],
}
visited =
set()
def dfs(visited, Mytree , node):
if node not in visited:
print
(node)
visited.add(node)
for neighbour in Mytree[node]:
dfs(visited, Mytree , neighbour)
dfs(visited, Mytree ,
'2')
```

Course Name: AIML Lab

Course Code: 21CSH-316

#### 4. Output:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

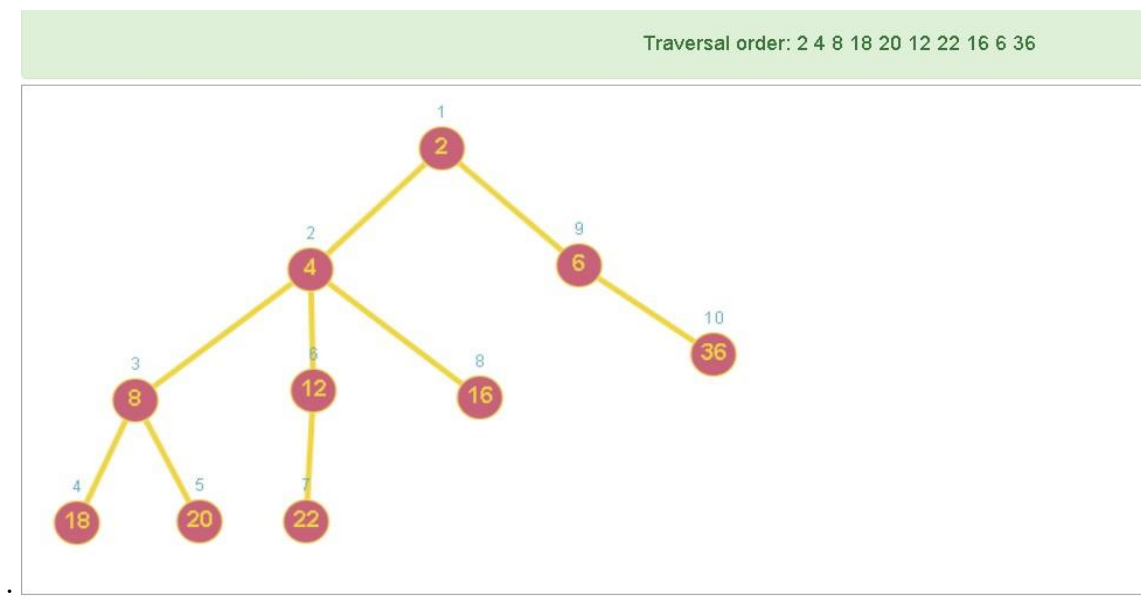
```
PS C:\Users\lenovo\Desktop\Coding> python -u "c:\Users\lenovo\Desktop\Coding\practical.py"
2
4
8
18
20
12
22
16
6
36
PS C:\Users\lenovo\Desktop\Coding>
```

#### 5. Learning and Outcomes:

In this experiment we have learnt about depth first search which uses stack to keep the track of nodes to be visited and revisited during the traversal process. We start from choosing a starting node and mark the node as visited and explore neighbors then recurse into neighbors , following by backtrack to previous nodes when no other nodes are present. Repeating the process till every node is visited.

On paper we learn and implement how to use stack and pop method and create final Visited list.

Mytree graph representation ; verify at : <http://graphonline.ru/en/?graph=tEnizAjDrrxTMbEp>



Name: UTKARSH JOSHI

UID: 21BCS9158