**Experiment: 1.2**

**Student Name:** SANJIV GUPTA **UID:** 21BCS-*3478*

**Branch:** CSE **Section/Group:** 21BCS-IOT-602B

**Semester:** 5th **Date:** 25/08/23

**Subject Name**: AIML Lab **Subject Code:** 21CSH-316

1. **AIM:** *Implement the DFS algorithm and analyze its performance and characteristics*
2. **Objective:** *The objective of this experiment is to implement the Depth-First Search (DFS) algorithm and analyze its performance and characteristics.*
3. **Tools/Resource Used:**

*1. Python programming language.*

*2. VS Code.*

1. **Algorithm:**

*1).Choose a node as the starting point for traversal.*

*2).Create an empty set to track visited nodes.*

*3).DFS Function:*

* *If the current node is not visited:*
* *Mark the current node as visited.*
* *Print its value.*
* *For each unvisited neighbor:*
* *Recursively perform DFS on the neighbor.*

*4).Begin DFS from the chosen starting node using the DFS function.*

*5).Explore as deeply as possible before backtracking to unexplored neighbors.*

*6).Stop when all reachable nodes are visited.*

1. **Program Code:**

*def dfs(graph, node, visited):*

*if node not in visited:*

*print(node, end=" ")*

*visited.add(node)*

*neighbors = graph[node]*

*for neighbor in neighbors:*

*dfs(graph, neighbor, visited)*

*graph = {*

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

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

*'C': ['F'],*

*'D': [],*

*'E': ['F'],*

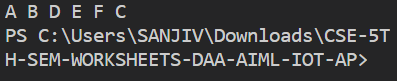
*'F': []*

*}*

*visited = set()*

*dfs(graph, 'A', visited)*

1. **Output/Result:**

****

1. **Learning Outcomes:**
2. *Implement a depth-first traversal (DFS) algorithm on a graph data structure.*
3. *Understand the concept of graph traversal and its importance in various applications.*
4. *Use recursion effectively to navigate through graph nodes and explore their connections.*