**EXPERIMENT 5 - Depth First Search**

**Aim:**

Write a python program to implement Depth First Search.

**Theoretical Background:**

Depth First Search or Depth First Traversal is an algorithm to look at all the vertices of a graph or tree data structure. The Depth First Search is a recursive algorithm that uses the concept of backtracking. It involves thorough searches of all the nodes by going ahead if potential, else by backtracking. Here, the word backtrack means, once you are moving forward and there are no more nodes along the present path, you progress backward on an equivalent path to seek out nodes to traverse. All the nodes are progressing to be visited on the current path until all the unvisited nodes are traversed after which subsequent paths are going to be selected.

The recursive method of the Depth First Search algorithm is implemented using stack. A standard Depth First Search implementation puts every vertex of the graph into one of the two categories (i) Visited and (ii) Not Visited. The only purpose of the algorithm is to visit all the vertices of the graph avoiding cycles.

**Algorithm:**

Step 1: Start.

Step 2: Read a graph from the user in the form of a dictionary where the parent node value is stored as the key value and corresponding child node value is stored as the pair of the key value. Store the value into the variable ‘graph’.

Step 3: Create a set for storing the value of the visited nods to keep track of the visited nodes of the graph.

Step 4: Create a function ‘dfs’ with the parameters as visited nodes, the graph and the node. Within the function perform the following:

Step 4.1: Start by putting any one of the graph’s vertices on top of the stack.

Step 4.2: After that take the top item of the stack and add it to the visited list of the vertex.

Step 4.3: Create a list of adjacent nodes of the vertex. Add the ones which aren’t in the visited list of the vertices to the top of the stack.

Step 4.4: Inside the function, we will check if any node of the graph is visited or not using the if condition. If it is not visited then print the node and add it to the visited set of nodes.

Step 4.5: Go to the neighboring node of the graph and again call the ‘dfs’ function to use the neighbor parameter.

Step 4.6: Continue from Step 4.2 to Step 4.5, until the stack is empty.

Step 5: Display the final result of the DFS by calling the ‘dfs’ function for the first time with the starting vertex of the graph.

Step 6: Stop.

**Conclusion:**

Python program to implement Depth First Search is successfully implemented.