THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXPERIMENT NO. 2.

Aim: Implement DFS search algorithm in python.

Theory:

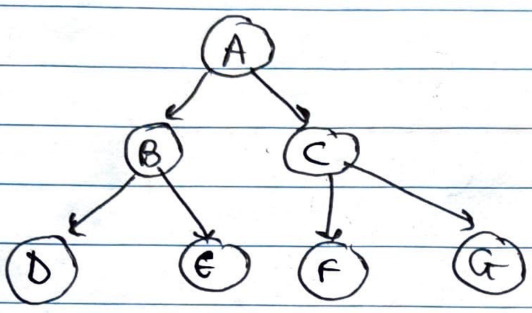
Depth first search or Depth first traversal is a recursive algorithm for searching all the vertices of a graph or tree data structure. The algorithm starts at root node & expands as far as possible along each branch before backtracking. It is a recursive algorithm to traverse & search. A DFS implementation puts each vertex into one of two categories: a) visited b) Not visited.
The marking of each vertex is done so as to avoid cycles.

1. Start by putting any one of the graph's vertices on top of a stack.
2. Take the top item of the stack & add it to the visited list.
3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the top of the stack.
4. Keep repeating steps 2 & 3 until the stack is empty.

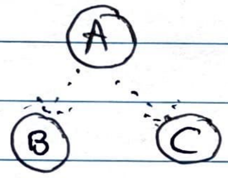
Applications of DFS algorithm:

1. For finding the path.
2. For detecting cycles in a graph.
3. For finding the strongly connected components of a graph.

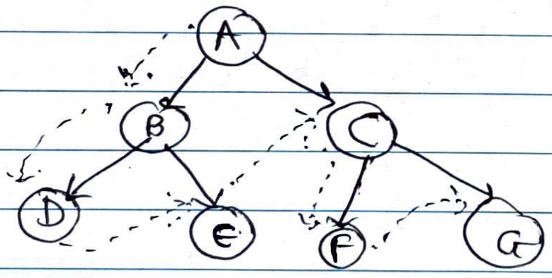
Example:



- starting from root node 'A'.



there are two available branches, considering $\textcircled{A} \to \textcircled{B}$ traversing till the depth of the branch



- A

$\therefore$ Path from root node 'A' is

$\textcircled{A} \to \textcircled{B} \to \textcircled{D} \to \textcircled{E} \to \textcircled{C} \to \textcircled{F} \to \textcircled{G}$

[illegible] 20/03/24

The time complexity of DFS algorithm is represented as O(V+E), where V is the no. of nodes & E is the no. of edges.
Space complexity of the algorithm is O(V).