

Implementing DFS using stack

```
class Solution {
public:
    // Function to return a list containing the DFS traversal of the graph.
    vector<int> dfsOfGraph(vector<vector<int>>& adj) {
        int V = adj.size(); // Number of vertices
        vector<int> dfsResult;
        vector<bool> visited(V, false);
        stack<int> st;

        st.push(0);

        while (!st.empty()) {
            int node = st.top();
            st.pop();

            if (!visited[node]) {
                visited[node] = true;
                dfsResult.push_back(node);

                // Push adjacent nodes in reverse order to maintain correct order
                for (auto it = adj[node].rbegin(); it != adj[node].rend(); ++it) {
                    if (!visited[*it]) {
                        st.push(*it);
                    }
                }
            }
        }
        return dfsResult;
    }
};
```

32% Solved

Courses ▾Tutorials ▾Jobs ▾Practice ▾Contests ▾

⌵

ProblemEditorialSubmissionsComments

Output Window

Compilation Results

Custom InputY.O.G.I. (AI Bot)

Problem Solved Successfully ✓

Suggest Feedback

Test Cases Passed

1120 / 1120

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored ⓘ

2 / 2

Your Total Score: 32 ↑

Time Taken

0.06

Solve Next

BFS of graphMother VertexCount the paths

C++ (g++ 5.4)

Start Timer

1 // } Driver Code Ends
2
3
4
5 class Solution {
6 public:
7 // Function to return a list containing the DFS traversal of the graph.
8 vector<int> dfsOfGraph(vector<vector<int>>& adj) {
9 int V = adj.size(); // Number of vertices
10 vector<int> dfsResult;
11 vector<bool> visited(V, false);
12 stack<int> st;
13
14 st.push(0);
15
16
17
18 while (!st.empty()) {
19 int node = st.top();
20 st.pop();
21
22
23 if (!visited[node]) {
24 visited[node] = true;
25 dfsResult.push_back(node);
26
27 // Push adjacent nodes in reverse order to maintain correct order
28 for (auto it = adj[node].rbegin(); it != adj[node].rend(); ++it) {
29 if (!visited[*it]) {
30 st.push(*it);
31 }
32 }
33 }
34 }
35 return dfsResult;
36 }
37 }
38
39
40 // } Driver Code Ends

Ctrl + Enter