BFS of graph

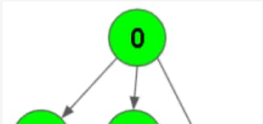
Easy | Accuracy: 44.09% | Submissions: 242K+ | Points: 2

Given a directed graph. The task is to do Breadth First Traversal of this graph starting from 0.

Note: One can move from node u to node v only if there's an edge from u to v and find the BFS traversal of the graph starting from the 0th vertex, from left to right according to the graph. Also, you should only take nodes directly or indirectly connected from Node 0 in consideration.

Example 1:

Input:



C++ (g++ 5.4) | Average Time: 10m | Start Timer

```cpp
// } Driver Code Ends
class Solution {
  public:
    // Function to return Breadth First Traversal of given graph.
    vector<int> bfsOfGraph(int V, vector<int> adj[]) {
        // Code here
        vector<int>res;
        vector<int>vis(V,0);
        queue<int>q;
        q.push(0);

        vis[0] =1;

        while(!q.empty()){
            int node = q.front();
            q.pop();
            res.push_back(node);

            for(int it:adj[node]){
                if(!vis[it]){
                    q.push(it);
                    vis[it] =1;
                }
            }
        }
        return res;
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit

DFS of Graph

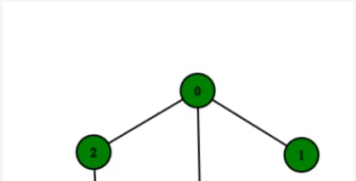
Easy Accuracy: 63.07% Submissions: 157K+ Points: 2

You are given a connected undirected graph. Perform a Depth First Traversal of the graph.

Note: Use a recursive approach to find the DFS traversal of the graph starting from the 0th vertex from left to right according to the graph.

Example 1:

Input: V = 5 , adj = [[2,3,1] , [0], [0,4], [0], [2]]



C++ (g++ 5.4) Average Time: 5m Start Timer

```cpp
// } Driver Code Ends
class Solution {
  public:
    // Function to return a list containing the DFS traversal of the graph.
   void dfs(int node,vector<int>&res,vector<int>&visited,vector<int>adj[]){
        res.push_back(node);

        visited[node] = 1;
        for(auto it:adj[node]){
            if(visited[it]==0)
            dfs(it,res,visited,adj);
        }
    }
    vector<int> dfsOfGraph(int V, vector<int> adj[]) {
        // Code here
        vector<int>visited(V,0);
        vector<int>res;

        dfs(0,res,visited,adj);
        return res;
    }
};
// } Driver Code Ends
```

Custom Input | Compile & Run | Submit