**DFS Traversal**

#include <vector>

void dfs(int node, vector<int> adj[], vector<int>& vis, vector<int>& ds) {

vis[node] = 1;

ds.push\_back(node);

for (auto it : adj[node]) {

if (!vis[it]) {

dfs(it, adj, vis, ds);

}

}

}

vector<vector<int>> depthFirstSearch(int V, int E, vector<vector<int>>& edges) {

vector<int> adj[V];

for (int i = 0; i < E; i++) {

adj[edges[i][0]].push\_back(edges[i][1]);

adj[edges[i][1]].push\_back(edges[i][0]);

}

vector<int> vis(V, 0);

vector<vector<int>> ans;

for (int i = 0; i < V; i++) {

if (!vis[i]) {

vector<int> ds;

dfs(i, adj, vis, ds);

ans.push\_back(ds);

}

}

return ans;

}