HPC-1B(DFS)

#include <iostream>

#include <vector>

#include <omp.h>

Using namespace std;

// Simple Parallel DFS function

Void dfs(const vector<vector<int>>& graph, vector<bool>& visited, int node) {

    #pragma omp critical

    {

        If (visited[node]) return;

        Visited[node] = true;

        Cout << “Visited: “ << node << endl;

    }

    For (int neighbor : graph[node]) {

        #pragma omp task

        {

            If (!visited[neighbor]) {

                Dfs(graph, visited, neighbor);

            }

        }

    }

    #pragma omp taskwait

}

Int main() {

    Int nodes, edges;

    Cout << “Enter number of nodes: “;

    Cin >> nodes;

    Vector<vector<int>> graph(nodes);

    Vector<bool> visited(nodes, false);

    Cout << “Enter number of edges: “;

    Cin >> edges;

    Cout << “Enter edges (u v):” << endl;

    For (int I = 0; I < edges; i++) {

        Int u, v;

        Cin >> u >> v;

        Graph[u].push\_back(v);

        Graph[v].push\_back(u); // undirected

    }

    Int start;

    Cout << “Enter starting node for DFS: “;

    Cin >> start;

    Cout << “\nParallel DFS starting from node “ << start << “:\n”;

    #pragma omp parallel

    {

        #pragma omp single

        Dfs(graph, visited, start);

    }

    Return 0;

}