**Parallel DFS**

#include <iostream>

#include <vector>

#include <omp.h>

using namespace std;

vector<int> graph[100000];

bool visited[100000];

void dfs(int node) {

#pragma omp critical

if (visited[node])

return;

visited[node] = true;

cout << node << " ";

#pragma omp parallel for

for (int i = 0; i < graph[node].size(); i++) {

int next = graph[node][i];

#pragma omp task

dfs(next);

}

}

int main() {

int n, m, start;

cout << "Enter No of Nodes, Edges, and Start Node: ";

cin >> n >> m >> start;

cout << "Enter Pairs of Edges:\n";

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

int u, v; cin >> u >> v;

graph[u].push\_back(v);

graph[v].push\_back(u);

}

fill(visited, visited + n, false);

cout << "Parallel DFS Traversal: ";

#pragma omp parallel

{

#pragma omp single

dfs(start);

}

return 0;

}