Assignment 1b : DFS

#include <iostream>

#include <vector>

#include <stack>

#include <omp.h>

using namespace std;

void dfs(vector<vector<int>>& graph, int start,

vector<bool>& visited) {

stack<int> s;

s.push(start);

visited[start] = true;

#pragma omp parallel {

#pragma omp single {

while (!s.empty()) {

int vertex = s.top();

s.pop();

#pragma omp task firstprivate(vertex) {

for (int neighbor : graph[vertex]) {

if (!visited[neighbor]) {

s.push(neighbor);

visited[neighbor] = true;

#pragma omp task

dfs(graph, neighbor, visited);

}

}

}

}

}

}

void parallel\_dfs(vector<vector<int>>& graph, int start) {

vector<bool> visited(graph.size(), false);

dfs(graph, start, visited);

}