Assignment 1: bfs

#include <iostream>

#include <vector>

#include <queue>

#include <omp.h>

using namespace std;

void bfs(vector<vector<int>>& graph, int start, vector<bool>&

visited) {

queue<int> q;

q.push(start);

visited[start] = true;

#pragma omp parallel {

#pragma omp single {

while (!q.empty()) {

int vertex = q.front();

q.pop();

#pragma omp task firstprivate(vertex) {

for (int neighbor : graph[vertex]) {

if (!visited[neighbor]) {

q.push(neighbor);

visited[neighbor] = true;

#pragma omp task

bfs(graph, neighbor, visited);

}

}

}

}

}

}

}

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

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

bfs(graph, start, visited);

}