ЗВІТ

про виконання лабораторної роботи № 9

Характеристики графів

з дисципліни «Обхід графа пошуком вшир»

студента групи КН-2226Б

Медвідь Дмитро Степанович

Варіант 11

Завдання №1

1. Умова завдання;

Реалізувати програмне застосування (програму), яке виконує наступні функції. Причому на вхід програми подається вхідний файл з описом графу, зі структурою, яка вказана у практичному завданні «Представлення графів». При реалізації алгоритмів вважати, що заданий граф є зв’язаним.

Обійти граф пошуком вшир. Користувач вводить початкову вершину графу. Програма виконує обхід графу, починаючи з вказаної початкової вершини. На екран виводиться протокол обходу – таблиця, яка містить наступні дані по кожній ітерації алгоритму обходу: поточна вершина, її BFS-номер, вміст черги.

1. текст програми;

const fs = require('fs');

// Функція для зчитування графу з файлу

function readGraphFromFile(filename) {

    const data = fs.readFileSync(filename, 'utf8');

    const lines = data.trim().split('\n');

    const vertices = parseInt(lines[0]);

    const edges = parseInt(lines[1]);

    let adjacencyList = Array.from({ length: vertices }, () => []);

    for (let i = 2; i < edges + 2; i++) {

        const [start, end] = lines[i].trim().split(' ').map(Number);

        adjacencyList[start].push(end);

    }

    return adjacencyList;

}

// Функція для обходу графу пошуком у ширину

function BFS(graph, startVertex) {

    let visited = Array(graph.length).fill(false);

    let queue = [];

    let protocol = [];

    queue.push(startVertex);

    while (queue.length) {

        let currentVertex = queue.shift();

        if (!visited[currentVertex]) {

            visited[currentVertex] = true;

            protocol.push({

                vertex: currentVertex,

                bfsNumber: protocol.length + 1,

                queueContents: [...queue]

            });

            for (let neighbor of graph[currentVertex]) {

                if (!visited[neighbor]) {

                    queue.push(neighbor);

                }

            }

        }

    }

    return protocol;

}

// Головна функція

function main() {

    const graph = readGraphFromFile('./lab9/graph.txt');

    const startVertex = 0; // Початкова вершина для обходу

    const protocol = BFS(graph, startVertex);

    console.log('Протокол обходу:');

    console.log('Вершина\tBFS-номер\tВміст черги');

    for (let entry of protocol) {

        console.log(`${entry.vertex}\t${entry.bfsNumber}\t\t${entry.queueContents.join(', ')}`);

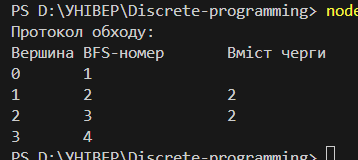
    }

}

// Виклик головної функції

main();

1. значення вхідних даних та результати виконання програми;



1. посилання на публічний репозиторій GitHub з кодом програми;

https://github.com/DartVeyder/Discrete-programming/blob/main/lab9/task1.js

1. Висновок

Дослідив роботу алгоритмів обходу в графах: пошуку вшир.