ЗВІТ

про виконання лабораторної роботи № 9

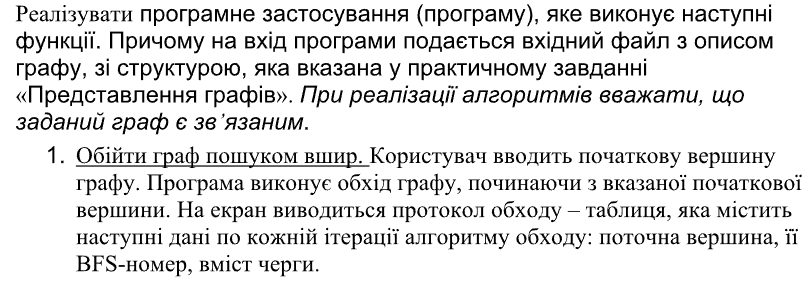
*Обхід графа пошуком вшир*

з дисципліни «Програмування дискретних структур»

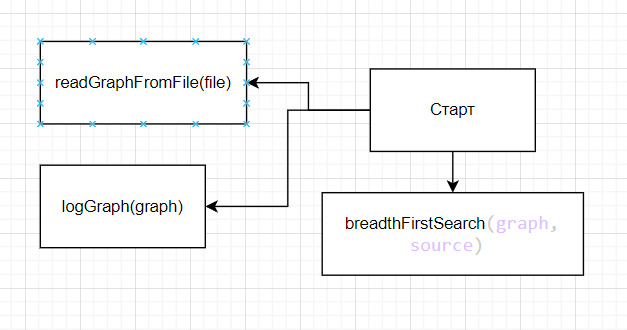
студента групи КН-2226

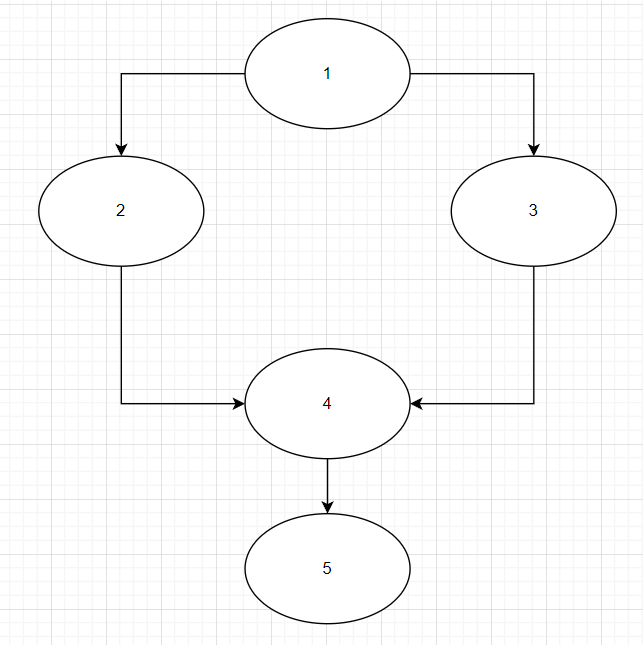
*Веждел Андрій Іванович*

**Умова завдання:**



**Блок схема:**

****

**Граф:**

**Текст коду:**

const fs = require("fs");

function readGraphFromFile(filename) {

  try {

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

    const lines = data.split("\n");

    const [n, m] = lines[0].split(" ").map(Number);

    const list = {};

    for (let i = 1; i < lines.length; i++) {

      let [v, u] = lines[i].split(" ").map(Number);

      if (!u || undefined) {

        u = [];

      }

      if (!list[v]) {

        list[v] = [];

      }

      list[v].push(u);

    }

    return { n, m, list };

  } catch (error) {

    console.error("Помилка при читанні файлу:", error.message);

    return null;

  }

}

function logGraph(graph) {

  const list = graph.list;

  for (const vertex in list) {

    const neighbors = list[vertex];

    neighbors.forEach((neighbor) => {

        console.log(` ${vertex} --> ${neighbor}`);

      }

    );

  }

}

function breadthFirstSearch(graph, source) {

  const queue = [source];

  while(queue.length > 0){

    const curr = queue.shift();

    BFS\_queue.push(curr);

    console.log(curr);

    for (let neighbor of graph[curr]) {

        queue.push(neighbor);

        if (neighbor.length === 0) {

            return;

          }

    }

  }

}

let graph = readGraphFromFile("./lb9/lb9.txt");

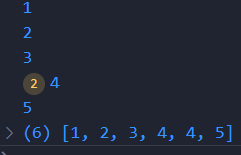
logGraph(graph);

let BFS\_queue = [];

breadthFirstSearch(graph.list, 1);

console.log(BFS\_queue);

**Результат виконання:**

****

**Посилання на гіт-репозиторій:**

[disc-systems/lb9/lb9.js at main · 1duxa/disc-systems (github.com)](https://github.com/1duxa/disc-systems/blob/main/lb2.js)

**Результат юніт-тесту:**

const fs = require("fs");

const { readGraphFromFile, breadthFirstSearch } = require("./lb9");

test("BFS traversal starting from vertex 1", () => {

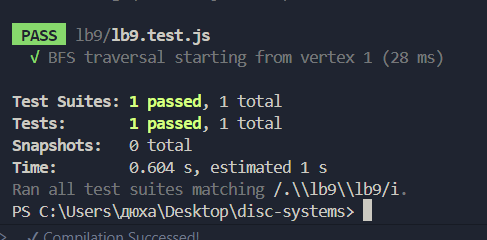
    const graph = readGraphFromFile("./lb9/lb9.txt");

    const expectedTraversalOrder = [1, 2, 3, 4, 4 ,5];

    const BFS\_stack=breadthFirstSearch(graph.list, 1);

    expect(BFS\_stack).toEqual(expectedTraversalOrder);

  });

****

**Висновок:**

Дослідив роботу алгоритмів обходу в графах: пошуку вшир.