ЗВІТ

про виконання лабораторної роботи № 10

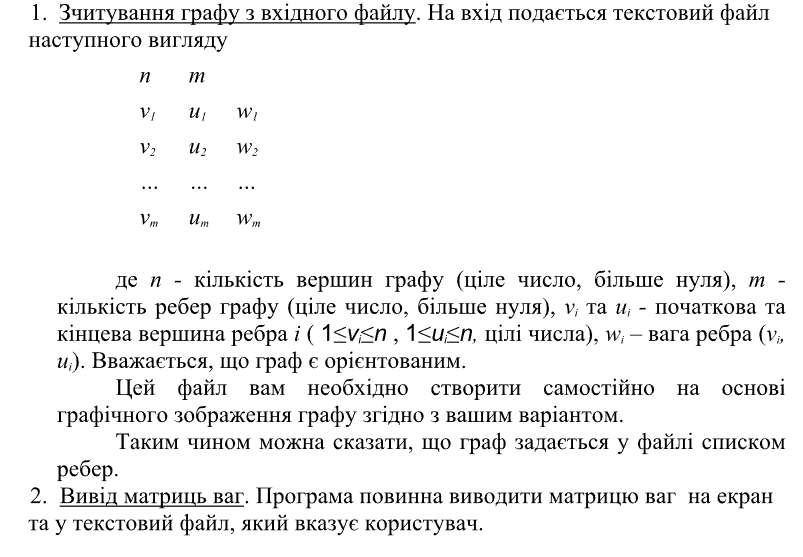
*Зважені графи*

з дисципліни «Програмування дискретних структур»

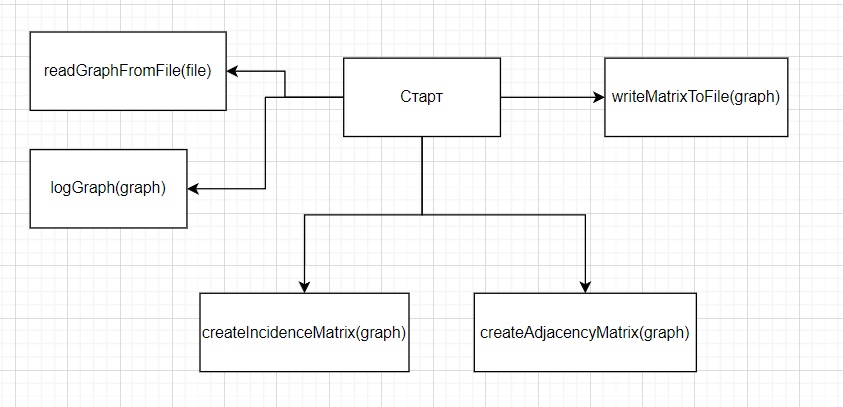
студента групи КН-2226

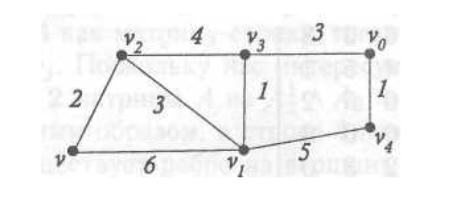
*Веждел Андрій Іванович*

**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, w] = lines[i].split(" ");

      if (!u || u === undefined) {

        u = [];

      }

      if (!list[v]) {

        list[v] = [];

      }

      list[v].push({ end:u, w: Number(w) });

    }

    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 key in list) {

    const neighbors = list[key];

    neighbors.forEach((neighbor) => {

      console.log(` ${key} --> ${neighbor.end} (вага: ${neighbor.w})`);

    });

  }

}

function createWeightMatrix(graph) {

    let { n, list } = graph;

    const weightMatrix = Array.from({ length: n }, () => Array(n).fill(Infinity));

    let count\_row = 0;

    for (let i = 0; i < n; i++) {

        weightMatrix[i][i] = 0;

    }

    for (let row\_key in list) {

      let count\_column = 0;

      for (let column\_key in list) {

        if (list[row\_key].some((item) => item.end === column\_key)) {

            weightMatrix[count\_row][count\_column] = list[row\_key].find((item) => item.end === column\_key).w;

        }

        count\_column++;

      }

      count\_row++;

    }

    return weightMatrix;

  }

  function writeMatrixToFile(matrix, filename) {

    const data = matrix.map(row => row.map(cell => cell.toString().padEnd(10, ' ')).join(" ")).join("\n");

    fs.writeFileSync(filename, data, "utf8");

    console.log(`Матрицю записано у файл: ${filename}`);

  }

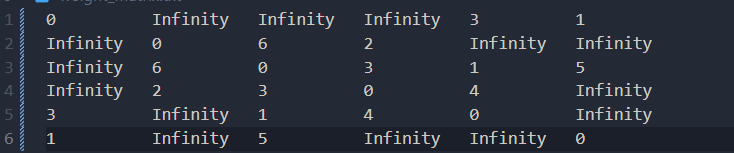
const graph = readGraphFromFile("./lb10/lb10.txt");

let matrix = createWeightMatrix(graph);

logGraph(graph);

writeMatrixToFile(matrix, "./lb10/weight\_matrix.txt");

**Результат виконання:**

****

**Посилання на гіт-репозиторій:**

[disc-systems/lb10/lb10.js at main · 1duxa/disc-systems (github.com)](https://github.com/1duxa/disc-systems/blob/main/lb2.js)

**Результат юніт-тесту:**

const lb6 = require("./lb6");

test("Матриця інцидентності", () => {

    const incidenceMatrix = [[-1,-1,0,0,0,1],[1,0,-1,0,0,0],[0,1,0,-1,0,0],[0,0,1,1,-1,0],[0,0,0,0,1,-1]];

    const graph = {

        n: 5,

        m: 6,

        list: {

            "1": ["2", "3"],

            "2": ["4"],

            "3": ["4"],

            "4": ["5"],

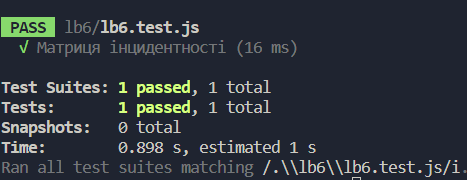
            "5": ["1"],

        },

    };

    expect(lb6.createIncidenceMatrix(graph)).toEqual(incidenceMatrix);

});

****

**Висновок:**

Розглянув та вивчити різні типи представлення зважених графів в пам’яті

комп`ютерів.