ЗВІТ

про виконання лабораторної роботи № 5

Множини

з дисципліни «Програмування дискретних структур»

студента групи КН-2226Б

Медвідь Дмитро Степанович

Варіант 11

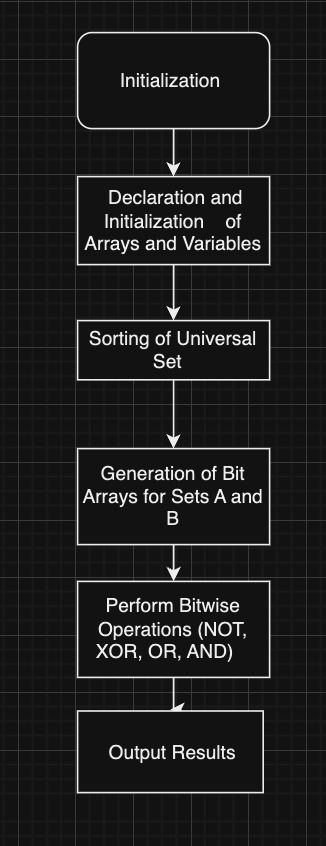
Завдання №1

1. Умова завдання;

Скласти програми із зазначеними вхідними даними та результатами.

1. Дано скінчені множини А={a, b, c}, В={x, y, z}, С={0, 1}. Побудувати множину ВxСxВ

1. опис алгоритму (блок-схема або структурна схема);



1. текст програми;

function buildD(A, B, C) {

    let D = [];

    let j = 0;

    for (let iC = 0; iC < C.length; iC++) {

        for (let iA = 0; iA < A.length; iA++) {

            for (let iB = 0; iB < B.length; iB++) {

                D[j] = C[iC] + A[iA] + B[iB];

                j++;

            }

        }

    }

    return D;

}

let A = ['a', 'b', 'c'];

let B = ['x', 'y', 'z'];

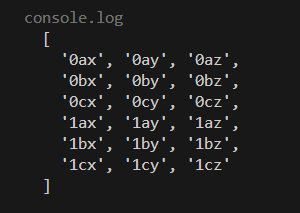
let C = ['0', '1'];

let D = buildD(A, B, C);

console.log(D);

module.exports = buildD;

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/lab5/task1.js

1. результати unit-тесту

// index.test.js

const buildD = require('./task1');

test('buildD builds the correct array', () => {

    const A = ['a', 'b', 'c'];

    const B = ['x', 'y', 'z'];

    const C = ['0', '1'];

    const result = buildD(A, B, C);

    expect(result).toEqual([

        '0ax', '0ay', '0az',

        '0bx', '0by', '0bz',

        '0cx', '0cy', '0cz',

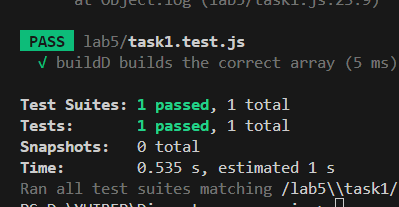
        '1ax', '1ay', '1az',

        '1bx', '1by', '1bz',

        '1cx', '1cy', '1cz'

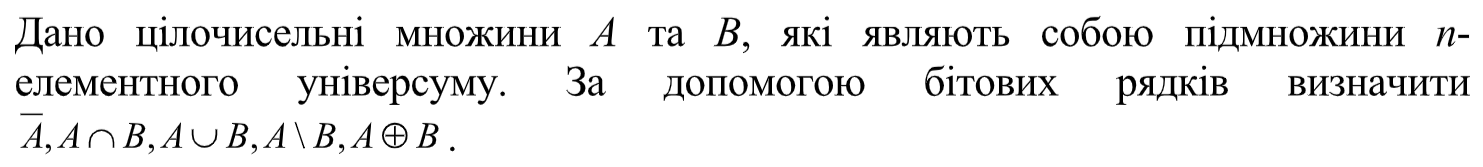
    ]);

});



Завдання №2

1. Умова завдання;



1. Опис алгоритму (блок-схема або структурна схема);
2. текст програми;

function bitwiseSetOperations(U, A, B) {

    // Sort the elements of the universe in ascending order

    U.sort((a, b) => a - b);

    // Initialize bitsets for A and B

    const Abit = new Array(U.length).fill(0);

    const Bbit = new Array(U.length).fill(0);

    // Create bitsets for A and B

    for (let i = 0; i < U.length; i++) {

        for (let j = 0; j < A.length; j++) {

            if (U[i] === A[j]) {

                Abit[i] = 1;

                break;

            }

        }

        for (let j = 0; j < B.length; j++) {

            if (U[i] === B[j]) {

                Bbit[i] = 1;

                break;

            }

        }

    }

    // Calculate bitsets for intersection, union, difference, and XOR

    const AperB = Abit.map((a, i) => a & Bbit[i]);

    const AobB = Abit.map((a, i) => a | Bbit[i]);

    const ArizB = Abit.map((a, i) => a & ~Bbit[i]);

    const AxorB = Abit.map((a, i) => (a ^ Bbit[i]) & 1);

    // Print results

    console.log("Bitset of A:", Abit.join(""));

    console.log("Bitset of B:", Bbit.join(""));

    console.log("Bitset of A ∩ B:", AperB.join(""));

    console.log("Bitset of A ∪ B:", AobB.join(""));

    console.log("Bitset of A \\ B:", ArizB.join(""));

    console.log("Bitset of A ⊕ B:", AxorB.join(""));

    // Return the result object

    return { Abit, Bbit, AperB, AobB, ArizB, AxorB };

}

// Example usage

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

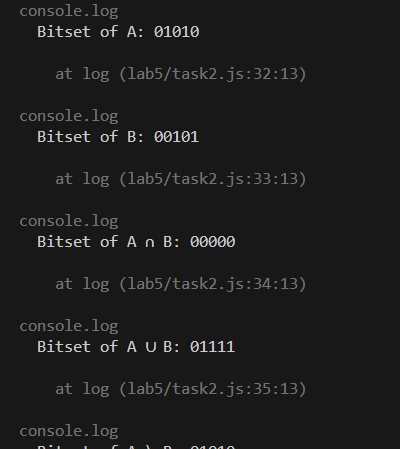
const A = [2, 4];

const B = [3, 5];

bitwiseSetOperations(U, A, B);

module.exports = bitwiseSetOperations;

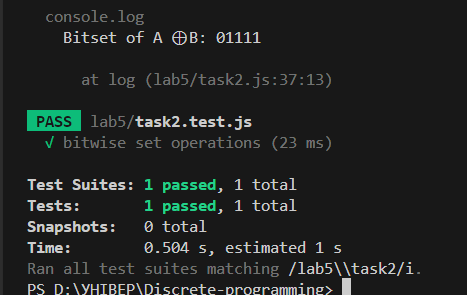
1. значення вхідних даних та результати виконання програми;



1. посилання на публічний репозиторій GitHub з кодом програми;

https://github.com/DartVeyder/Discrete-programming/blob/main/lab5/task2.js

1. результати unit-тесту



Висновок я Поглибив і закріпив розуміння положень теорії множин, формувати навички складання алгоритмів та програм опрацювання множин.