Министерство Образования Республики Беларусь

Учреждение образования

«Гомельский государственный технический университет

имени П.О.Сухого»

Кафедра «Информатика»

**Лабораторная работа № 2**

по дисциплине: **«Системный анализ и исследование операций»**

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

Выполнил студент

группы ИП-41

Коваленко А.И.

Проверил преподаватель

Бородин Н.Н.

Гомель 2023

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Цель работы:** получить практические навыки решения задач целочисленного программирования методом ветвей и границ.

**Задание**:

С помощью средств Visual Studio применяя метод ветвей и границ, решить задачу коммивояжера и построить полное дерево ветвления с матрицей расстояний, заданной таблицей.



Ход выполнения:

let array = [

    [-1, 48, 27, 31, 43],

    [33, -1, 28, 44, 43],

    [41, 28, -1, 40, 36],

    [37, 35, 29, -1, 46],

    [48, 48, 25, 29, -1]

];

function lab2(arr) {

    let r = 0;

    let rPath = [];

    for (let i = 0; i < arr.length - 1; i++) {

        r += arr[i][i + 1];

        rPath.push(i);

    }

    r += arr[arr.length - 1][0];

    let S = [{

        path: [0],

        array: JSON.parse(JSON.stringify(arr)),

        score: getScore(arr)

    }];

    while (S.length > 0) {

        let min = S[0].score;

        let min\_i = 0;

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

            if (S[i].score < min) {

                min = S[i].score;

                min\_i = i;

            }

        let temp = JSON.parse(JSON.stringify(S[min\_i]));

        S.splice(min\_i, 1);

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

            if (temp.path.indexOf(i) == -1) {

                S.unshift({

                    path: temp.path.concat([i]),

                    array: removePath(arr, temp.path.concat([i])),

                    score: getScore(arr, temp.path.concat([i]))

                })

                if (S[0].score > r) S.shift();

                else if (S[0].path.length == arr.length - 1) {

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

                        if (S[0].path.indexOf(j) == -1) {

                            S[0].path.push(j);

                            j = arr.length;

                        }

                    S[0].path.push(0);

                    if (getLength(arr, S[0].path) < r) {

                        rPath = JSON.parse(JSON.stringify(S[0].path));

                        r = getLength(arr, S[0].path);

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

                            if (S[j].score > r) {

                                S.splice(j, 1);

                                j--;

                            }

                    }

                    S.shift();

                }

            }

        }

        console.log(S);

    }

    console.log(rPath);

    console.log(r);

    document.getElementById("result").innerHTML = rPath + "<br>" + r;

}

function remove(arr, i1, j1) {

    let a = JSON.parse(JSON.stringify(arr));

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

        a[i1][i] = -1;

        a[i][j1] = -1;

    }

    return a;

}

function removePath(arr, path) {

    let a = JSON.parse(JSON.stringify(arr));

    for (let i = 0; i < path.length - 1; i++)

        a = remove(a, path[i], path[i + 1]);

    return a;

}

function getScore(arr, path) {

    let a = JSON.parse(JSON.stringify(arr));

    let sum = 0;

    if (path != undefined)

        for (let i = 0; i < path.length - 1; i++) {

            sum += a[path[i]][path[i + 1]];

            a = remove(a, path[i], path[i + 1]);

        }

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

        let min = -1;

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

            if ((a[i][j] < min || min < 0) && a[i][j] >= 0) min = a[i][j];

        }

        if (min > 0) {

            sum += min;

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

                if (a[i][j] >= 0) a[i][j] -= min;

            }

        }

    }

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

        let min = -1;

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

            if ((a[j][i] < min || min < 0) && a[j][i] >= 0) min = a[j][i];

        }

        if (min > 0) {

            sum += min;

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

                if (a[j][i] >= 0) a[j][i] -= min;

            }

        }

    }

    return sum;

}

function getLength(a, path) {

    let length = 0;

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

        length += a[path[i]][path[i + 1]];

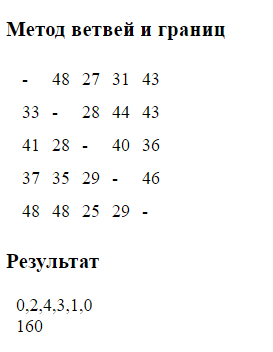
    }

    return length;

}

lab2(array);

Результат работы:



**Вывод**: получили практические навыки решения задач целочисленного программирования методом ветвей и границ.