МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем автоматизованого проектування



Звіт до лабораторної роботи №3

з дисципліни:

«Теорія прийняття рішень»

Варіант 14

Виконав:

студент групи КН-404

Коцур Н.О.

Прийняв:

Кривий Р.З.

Львів 2020

**Мета роботи**

Вивчення методів голосування і дослідження їх властивостей; опанування методикою обробки профілів колективного голосування, знаходження переможця та відновлення колективного ранжування.

**Порядок виконання роботи**

Задача.

Який кандидат виграє А,Б,С по методу Кондорсе і по методу Борда для заданого

розподілу голосів?

Порядок вирішення завдання:

1) Провести розрахунок для кожного методу колективного голосовування.

2) Описати порядок виконання роботи і вказати переможці за кожним методом

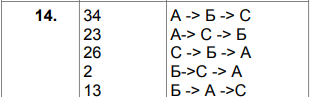
4) Реалізувати програмне забезпечення, яке б розв'язувало дану задачу. Мова

програмування неважлива. Обов'язково: дані мають зчитуватись з файлу і виводитись

у табличній формі.

**Завдання**

Варіант 14.



**Результати виконання**

1. ***Борда***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 34 | 23 | 26 | 2 | 13 | S |
| A | А | С | Б | Б | 2 |
| Б | C | Б | С | A | 1 |
| С | Б | А | А | C | 0 |

Переможець – кандидат A.

Колективне голосування: А>Б>C.

1. ***Кондорсе***

A:Б=57:41

A:C=70:28

Б:С=49:49

B:C=14:5

B:D=14:5

C:D=13:6

А виграє у Б та С, тому А є переможцем.

Колективне голосування: А>В=С.

Таблиця 1

|  |  |
| --- | --- |
| Правило голосування | Кандидат переможець |
| Правило Борда | A |
| Правило Кондорсе | А |

**Програмна реалізація**

const Max = (array) => Math.max(...array);

const max\_index = (array) => array.reduce((acc, cur, index) => (cur === Max(array) ? [...acc, index] : acc), []);

const has\_absolute\_win = (candidate) => candidate > ((sum(votes) / 2) + 1);

const grade = (num, size) => size - num - 1;

const candidates = [

    [1, 1, 3, 2, 2],

    [2, 3, 2, 3, 1],

    [3, 2, 1, 1, 3]

];

const votes = [34, 23, 26, 2, 13];

const candidates\_num = 3;

show\_results();

function show\_results() {

    console.log("Назва правила | Переможець");

    console.log("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

    // process.stdout.write("Відносна більшість                     | ");

    // show\_solutions(relative\_most(candidates, false));

    // process.stdout.write("Відносна більшість з вибуванням        | ");

    // show\_solutions(relative\_most(candidates, true));

    process.stdout.write("Борда                                  | ");

    show\_solutions(borda(candidates));

    process.stdout.write("Кондорсе                               | ");

    show\_solutions(condorse(candidates));

    // process.stdout.write("Копленда                               | ");

    // show\_solutions(coplend(candidates));

    // process.stdout.write("Сімпсона                               | ");

    // show\_solutions(simpson(candidates));

}

function show\_solutions(array) {

    array.forEach((elem, i) => {

        if (i == 0)

            process.stdout.write(1 + elem + " ");

    });

    console.log();

    console.log("‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾");

}

function relative\_most(candids, is\_absolute) {

    let uppers = candids[0];

    let upper\_indices = [];

    let results = [];

    for (let i = 0; i < candidates\_num; i++) {

        upper\_indices[i] = new Array();

    }

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

        upper\_indices[uppers[i] - 1].push(i);

    }

    for (let i = 0; i < candidates\_num; i++) {

        results[i] = votes\_sum(upper\_indices[i]);

    }

    if (!is\_absolute) {

        return max\_index(results);

    }    else {

        return max\_index(second\_tour\_winner(results));

    }

}

function second\_tour\_winner(first\_tour\_results) {

    let winner = max\_index(first\_tour\_results);

    let second = max\_index(first\_tour\_results);

    first\_tour\_results.sort((a, b) => b - a);

    let first\_tour\_winner = first\_tour\_results[0];

    if(has\_absolute\_win(first\_tour\_winner)) {

        return max\_index(first\_tour\_results)[0];

    } else {

        return first\_tour\_winner > sum(votes) - first\_tour\_winner ?

            winner : second;

    }

}

function borda(candids) {

    let results = new Array(candids.length).fill(0);

    candids.forEach((row, x) => {

        row.forEach((elem, y) => {

            results[elem - 1] += grade(x, candids.length) \* votes[y];

        });

    });

    return max\_index(results);

}

function condorse(candids) {

    let results = comparative\_votes(candids);

    return condorse\_winner(results);

}

function coplend(candids) {

    let results = comparative\_votes(candids);

    let most = (sum(votes)) / 2;

    results.forEach((row, i) => {

        row.forEach((elem, j) => {

            results[i][j] = elem > most ? 1 : -1;

        });

    });

    let sums = coplend\_sums(results);

    return new Array(1).fill(2);

}

function simpson(candids) {

    let results = comparative\_votes(candids);

    let grades = [];

    results.forEach((row) => {

        grades.push(min(row));

    });

    return new Array(1).fill(2);

}

function coplend\_sums(results) {

    let sums = [];

    results.forEach((row) => {

        sums.push(sum(row));

    });

    return sums;

}

function comparative\_votes(candids) {

    let results = new Array(5).fill(new Array(3).fill(0));

    let transposed\_candidates = transposed(candids);

    transposed\_candidates.forEach((row, i) => {

        row.forEach((\_, j) => {

            for (let ind = 0; ind < votes.length; ind++) {

                let cand = transposed\_candidates[i].indexOf((j + 1));

                results[i][j] += votes[i] \* is\_greater(cand, transposed\_candidates[ind].indexOf((j + 1)));

            }

        });

    });

    return results;

}

function condorse\_winner(results) {

    let winner\_counter = 0;

    let winners = [];

    results.forEach((row, i) => {

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

            if (results[i][j] < results[j][i]) {

                break;

            } else {

                winner\_counter++;

            }

        }

        if (winner\_counter == 3) {

            winners.push(i);

        }

        winner\_counter = 0;

    });

    return (winners);

}

function is\_greater(cand, j) {

    return j > cand ? 1 : 0;

}

function transposed(matrix) {

    return matrix[0].map((\_, i) => matrix.map(row => row[i]));

}

function min(array) {

    let min = 100;

    array.forEach((elem) => {

        if (elem > 0)

            min = elem >= min ? min : elem;

    });

    return min;

}

function sum(array) {

    let sum = 0;

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

        sum += parseInt( array[i], 10 );

    }

    return sum;

}

function votes\_sum(array) {

    var result = 0;

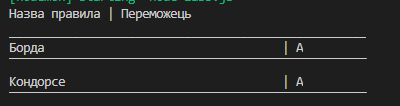
    array.forEach((elem) => {

        result += votes[elem];

    });

    return result;

}



**Висновок.** У цій лабораторній роботі я вивчив методи голосування і дослідив їх властивостіа також опанував методи голосування Борда та Кондорсе.