Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики і обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

З ООП

*Виконав:*

Студент групи ІО-34

Кривоносов О.O.

НЗК= 3431

*Перевірив:*

Симоненко В.П.

м. Київ

2016 р.

Варiант: 31%13 + 1 = 6

6. Реалізувати модель просторового планувальника на основі алгоритму перетворення матриці зв'язності з вибудовуванням "1" по головній діагоналі.

**Описание алгоритма**

1. Находим строку с минимальным количеством «1».

2. Меняем местами данную строка с первой.

3. Среди столбцов, которые в первой строке имеют значение «1» находим столбец с максимальным количеством «1».

4. Меняем местами столбец с первым.

5. Обрезаем первый столбец и первую строку.

6. Выполняем предыдущие действия для получения матрицы учитывая целостность общей.

Временная сложность для первой итерации О1(n)=2(n2+n)

Для i-ой Оi(n)=2((n-i)2+(n-i))

О(n)=2n(n2+n)=n3+n2

Листинг программы

/\*\*

\* Created by oleksii on 25.05.16.

\*/

var matrix = [];

function mainApp(){

var subMatrix = [];

for(var i = 0; i<matrix.length-1; i++){

subMatrix = sliceMatrix(matrix,i);

subMatrix = changeRow(subMatrix, searchMinRow(subMatrix));

subMatrix= changeCol(subMatrix,searchMinCol(subMatrix));

// subMatrix = fillZeros(subMatrix);

matrix = joinMatrix(matrix, i, subMatrix);

matrixToString(matrix);

}

matrixToViev(matrix, 'finish-matrix', 'finish');

}

function joinMatrix(matrix, n, subMatrix){

for(var i = 0; i < subMatrix.length; i++){

for(var j = 0; j < subMatrix.length; j++){

matrix[i+n][j+n] = subMatrix[i][j];

}

}

return matrix;

}

function sliceMatrix(matrix, n){

var resultMatrix = [];

for(var i = 0; i< matrix.length-n;i++){

resultMatrix[i] = [];

for(var j =0; j < matrix.length-n; j++){

resultMatrix[i][j] = matrix[i+n][j+n]

}

}

return resultMatrix;

}

function matrixToString(matrix){

var string = '';

matrix.forEach(function(el){

el.forEach(function(el2){

string+= el2 + ' ';

});

string +="\n"

});

console.log(string);

return string;

}

function searchMinRow(matrix){

var min = Infinity;

var sumResult;

var result = 0;

matrix.forEach(function(row, i){

sumResult = row.reduce(function(sum, current){

return sum+current;

},0);

console.log(sumResult,min);

if(sumResult < min && sumResult != 0){

result = i;

min = sumResult;

}

});

return result;

}

function searchMinCol(matrix){

var min = -Infinity;

var sumResult;

var result = 0;

for(var i = 0; i < matrix.length; i++){

if(matrix[0][i] == 0)

continue;

sumResult = 0;

for(var j = 0; j < matrix.length; j++){

sumResult+= matrix[j][i];

}

if(sumResult > min){

result = i;

min = sumResult;

}

}

return result;

}

function changeRow(matrix, n){

var buf = matrix[n];

matrix[n] = matrix[0];

matrix[0] = buf;

return matrix;

}

function changeCol(matrix, n){

var buf;

for(var i = 0; i < matrix.length; i++){

buf = matrix[i][n];

matrix[i][n] = matrix[i][0];

matrix[i][0] = buf;

}

return matrix;

}

function fillZeros(matrix){

for(var i = 1; i < matrix.length; i++){

matrix[0][i] = 0;

matrix[i][0] = 0;

}

return matrix;

}

function generator(){

matrix = [];

var n = $('#size').val();

for(var i = 0; i<n; i++){

matrix[i]=[];

for(var j = 0; j<n; j++){

matrix[i][j] = randomInteger(0,1);

}

}

matrixToViev(matrix, 'start-matrix', 'start');

}

function randomInteger(min, max) {

var rand = min - 0.5 + Math.random() \* (max - min + 1)

rand = Math.round(rand);

return rand;

}

function matrixToViev(matrix, apTo, curId){

$('#'+apTo+' tbody').empty();

for(var i= 0; i < matrix.length; i++){

$('#' + apTo + ' tbody').append('<tr id="r'+curId+i+'"></tr>')

for(var j = 0; j < matrix.length; j++){

$('#r'+curId+i).append('<td id="'+curId+i+j+'">'+matrix[i][j]+' </td>');

}

}

}