Министерство образования и науки РФ

Тамбовский государственный технический университет

Кафедра

Отчет по лабораторной работе №1

по дисциплине «Программирование»

Выполнил: студент группы

.

Проверил:.

Тамбов 20

1. Постановка задания:

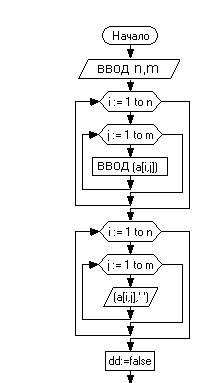
Дана матрица [n\*m]. Установить, имеются ли в ней седловые точки. Сортировать матрицу по возрастанию методом бинарных вставок. Проверить, сохранили ли седловые точки свой статус. Точка является седловой, если она min в строке и max в столбце.

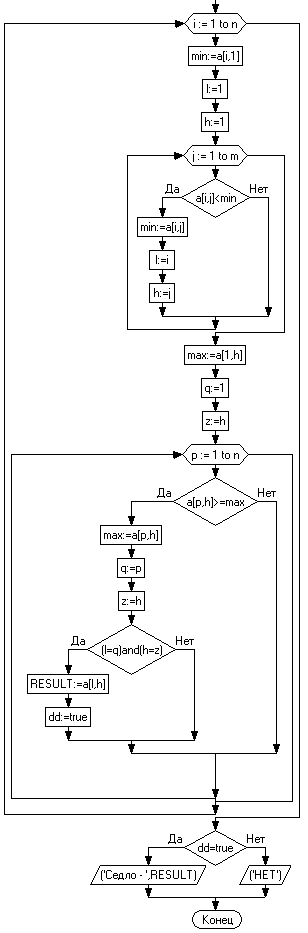
1. Краткое описание алгоритма:

Для решения поставленной задачи необходимо выполнить следующие шаги:

1. Создаем массив [n][m]
2. Заполняем массив рандомными числами от 1 до 100
3. Ищем седловые точки, выводим результат поиска
4. Переписываем двумерный массив в одномерный
5. Делаем бинарную сортировку одномерного массива
6. Переписываем одномерный массив в двумерный
7. Повторяем шаг 3
8. Принцип работы сортировки

Блок схема алгоритма:





1. Исходный текст программы:

#include <stdio.h>#include <stdlib.h>

#include <time.h> #include <conio.h>

int nN,mM;int schetchikA=0, schetchikB=0;

int \*\*arr; int \*odnomernMassivXY; int \*matrixA, \*matrixB;

int spMassA[3];int spMassB[3];

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void SaddlePoint()

{//1й поиск седл точ

int i,j, mincol;

matrixA[nN\*mM];//Храним значения точек

for (i=0;i<nN;i++)

{

mincol=0;

for(j=1; j<mM; j++)

if(arr[i][j]<arr[i][mincol]) mincol=j;

for(int y=0; y<mM; y++)

if(arr[i][y]==arr[i][mincol])

{

for(j=0; j<nN; j++) if(arr[j][y]>arr[i][mincol] && j!=i){break;}

if(j==nN)

{//Если сработал то есть точка [x,y]=a;

printf("Sedlovaya tochka: %d:%d //Znachenie %d\n",i,y,arr[i][y]);

spMassA[schetchikA]=arr[i][y];

schetchikA++;//Увеличиваем счетчикА

}

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int randomMassivArr()

{//Sozd rand massiv arr[][]

int i, j, n, m;

printf("Razmer matricy N\*M\n");

printf("Input size N: ");

scanf("%d", &n);

printf("Input size M: ");

scanf("%d", &m);

nN=n;mM=m;

arr = (int \*\*)malloc(n \* sizeof(int \*));

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

arr[i] = (int \*)malloc(m \* sizeof(int));

/\* Рандомизация/

srand(time(NULL));

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

for (j = 0; j < m; j++)

arr[i][j] = rand()% 13 + i;

\*/arr[0][0]=3;arr[0][1]=45;arr[0][2]=36;

arr[1][0]=6;arr[1][1]=16;arr[1][2]=26;

arr[2][0]=6;arr[2][1]=63;arr[2][2]=62;

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

{

for(j = 0; j < m; j++)

printf("%d \t", arr[i][j]);

printf("\n");

}

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void SortBinInsert(int \*odnomernMassivXY)

{//Na vxode polychem odnomernMassiv[]

for (int i = 1; i < nN\*mM; i++)

{

int low = 0;

int righ = i - 1;

int Temp = odnomernMassivXY[i];

while (low <= righ)

{//Find

int mid = (low + righ)/2;

if (Temp < odnomernMassivXY[mid])

righ = mid - 1;

else

low = mid + 1;

}

//backward shift

for (int j = i - 1; j >= low; j--)

odnomernMassivXY[j + 1] = odnomernMassivXY[j];

odnomernMassivXY[low] = Temp;

}

printf("\nOtsortirovanai massiv:");

for (int I=0; I< nN\*mM; I++)

printf("%d ",odnomernMassivXY[I]);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void convertArr\_in\_OdnomerMassiv()

{

odnomernMassivXY[nN\*mM];

int razmerMasivaXY = nN\*mM;

int n = 0, mm = 0;

for (int j = 0; j < razmerMasivaXY; j++) //(0; 0<16; 0++)

{

if (mm < mM) //0<=4

{

odnomernMassivXY[j] = arr[n][mm];//Zapolnaem

printf("%d ",odnomernMassivXY[j]);//Vevod elementa & probela

mm++; //0++

}

if (mm >= mM) //4>=4

{

n++; //0++

mm = 0;//Obnylaem [index in stroke]

}

}

SortBinInsert(odnomernMassivXY);//Vepolnaem metod sortirovki

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int odnomernMassivXY\_in\_DvumernMassivArr()

{

int i, j;

free(arr);

int value=0;

for (int ii = 0; ii < nN; ii++)

for (int jj = 0; jj < mM; jj++)

{

arr[ii][jj] = odnomernMassivXY[value];

value++;

}

for(i = 0; i < nN; i++)

{

for(j = 0; j < mM; j++)

printf("%d \t", arr[i][j]);

printf("\n");

}

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void SaddlePoint\_PosleSortirovki()

{//Poisk sedl tochk #2

matrixB[nN\*mM];//Храним значения точек

int i,j, mincol;

for (i=0;i<nN;i++)

{

mincol=0;

for(j=1; j<mM; j++)

if(arr[i][j]<arr[i][mincol])

mincol=j;

for(int y=0; y<mM; y++)

if(arr[i][y]==arr[i][mincol])

{

for(j=0; j<nN; j++)

if(arr[j][y]>arr[i][mincol] && j!=i)

{break;}

if(j==nN)

{

printf("Sedlovaya tochka: %d:%d //Znachenie %d\n",i,y,arr[i][y]);

spMassB[schetchikB]=arr[i][y];//keyB=arr[i][y]; matrixB[schetchikB]=keyB;

schetchikB++;

}

}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int ArraySize(int array[])

{

int i = 0;

while(array[i] != NULL) i++;

return i;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void StatusAB()

{int i,j;

if(schetchikA>=schetchikB)/\*3>=1\*/

{

for(j=0;j<schetchikB;j++)

for(i=0;i<schetchikA;i++){if(spMassA[i]==spMassB[j])printf(" %d]==[%d ",spMassA[i],spMassB[j]);

else printf(" %d]!=[%d ",spMassA[i],spMassB[j]);}

}else{/\*1<3\*/

for(i=0;i<schetchikA;i++)

for(j=0;j<schetchikB;j++)

{

if(spMassA[i]==spMassB[j])printf(" %d]==[%d ",spMassA[i],spMassB[j]);

else printf(" %d]==[%d ",spMassA[i],spMassB[j]);}

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void Garbedge()

{ //Osvob pamati//

for (int i = 0; i < nN; i++)free(arr[i]);

free(arr);free(odnomernMassivXY);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

void main()

{

randomMassivArr();

SaddlePoint();

printf("\nOdnomernai Massiv:\n");

convertArr\_in\_OdnomerMassiv();

printf("\nOtsortirovanay dvumernai massiv:\n");

odnomernMassivXY\_in\_DvumernMassivArr();

SaddlePoint\_PosleSortirovki();

StatusAB();

Garbedge();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//