НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №4

з предмету «Операційні системи»

Виконав   
студент гр. ІО-61, ФІОТ  
Алещенко Олексій Вадимович

Київ – 2009

1. **Завдання.**

Варіант №2

Скласти програму реалізації угорського методу для неоднорідної обчислювальної системи.

1. **Лістинг основної частини програми.**

// Программа реализует Венгерский алгоритм - "классика".

program VengrAgr;

{$APPTYPE CONSOLE}

uses

SysUtils;

const

R\_Max = 6; //максимальное к-во ресурсов

Z\_Max = 6; //максимальное к-во задач

type

TRVec = array [1..R\_Max] of integer;

TZVec = array [1..Z\_Max] of integer;

TMatrix = array [1..Z\_Max,1..R\_Max] of integer;

TMark = array [1..Z\_Max,1..2] of integer;

TBRVec = array [1..R\_Max] of boolean;

TBZVec = array [1..Z\_Max] of boolean;

var z,r: integer;

a: TMatrix;

nIter,nMZero: integer;

MarkedZero: TMark;

MarkZ: TBZVec;

MarkR: TBRVec;

procedure InPut;

var f:text;

i,j: integer;

begin

Assign(f,'input.txt');

reset(f);

readln(f,z,r);

for i:=1 to z do

for j:=1 to r do

read(f,a[i,j]);

close(f);

end;

//Отнимаем из каждого столбца минимальный эл-т (этого столбца).

procedure Step1;

var i,j: integer;

min: integer;

begin

for j:=1 to r do

begin

min:=2147483647;

for i:=1 to z do

if (a[i,j]<min)and(a[i,j]>=0) then

min:=a[i,j];

if min<>0 then

for i:=1 to z do

a[i,j]:=a[i,j]-min;

end;

end;

//Из полученной матрицы из каждой строки отнимаем минимальный эл-т (этой строки).

procedure Step2;

var i,j: integer;

min: integer;

begin

for i:=1 to z do

begin

min:=2147483647;

for j:=1 to r do

if (a[i,j]<min)and(a[i,j]>=0) then

min:=a[i,j];

if min<>0 then

for j:=1 to r do

a[i,j]:=a[i,j]-min;

end;

end;

//Из ноликов сделали 1цы и находим макс. паросочетание.

procedure Step3;

var i,j,ki,kj: integer;

mas: TMatrix;

aZ: TZVec;

aR: TRVec;

//Функции сортировки вектора\*

function ZSort(vec:TZVec):TZVec;

var index,jndex,max: integer;

begin

for index := 1 to z do

begin

max := 1;

for jndex := 1 to z do

begin

if vec[max]<vec[jndex] then

max := jndex;

end;

ZSort[index] := max;

vec[max] := 0;

end;

end;

function RSort(vec:TRVec):TRVec;

var index,jndex,max: integer;

begin

for index := 1 to r do

begin

max := 1;

for jndex := 1 to r do

begin

if vec[max]<vec[jndex] then

max := jndex;

end;

RSort[index] := max;

vec[max] := 0;

end;

end;

begin

for i:=1 to z do

for j:=1 to r do

if a[i,j]=0 then

mas[i,j]:=1

else

mas[i,j]:=0;

for i:=1 to z do

aZ[i]:=0;

for j:=1 to r do

aR[j]:=0;

for i:=1 to z do

for j:=1 to r do

begin

aZ[i] := aZ[i]+mas[i,j];

aR[j] := aR[j]+mas[i,j];

end;

aZ:=ZSort(aZ);

aR:=RSort(aR);

nIter:=nIter+1;

nMZero:=0;

Writeln(' ..:: Iteration ',nIter,' ::..');

for i:=z downto 1 do

begin

for j:=r downto 1 do

if mas[aZ[i],aR[j]]=1 then

begin

Writeln('i=',aZ[i],' j=',aR[j]);

nMZero := nMZero+1;

MarkedZero[nMZero,1]:=aZ[i]; // Запоменаем "помеченые

MarkedZero[nMZero,2]:=aR[j]; // нули" (паросочетание).

for ki := 1 to z do

mas[ki,aR[j]] := 0;

for kj := 1 to r do

mas[aZ[i],kj] := 0;

break;

end;

end;

end;

//"Хождение" по строкам и столбцам.

procedure Step4;

var i,j:integer;

sw:boolean;

min:integer;

//Функция проверяет "отмечен ли ноль" (входить ли данная пара в максимальное

//паросочетание).

function isMark(ki,kj:integer):boolean;

var i:integer;

res:boolean;

begin

res:=false;

for i:=1 to nMZero do

if (MarkedZero[i,1]=ki)and(MarkedZero[i,2]=kj) then

begin

res:=true;

break;

end;

isMark:=res;

end;

begin

//Ищем строки, где отсутсвуют помеченые нули (соответствующие элементы вектора

//MarkZ останутся "false").

for i:=1 to nMZero do

MarkZ[MarkedZero[i,1]] := true;

sw:=true;

while sw do

begin

sw:=false;

//Ищем столбцы, где есть непомеченный ноль в этой строке (соответствующие

//элементы вектора MarkR станут "true").

for i:=1 to z do

if not(MarkZ[i]) then

for j:=1 to r do

if (a[i,j] = 0)and not(isMark(i,j)) then

if not(MarkR[j]) then

begin

MarkR[j]:=true;

sw:=true;

end;

if not(sw) then

break;

sw:=false;

//Ищем по этим столбцам строки, где есть помеченный ноль (в этом столбце).

for j:=1 to r do

if MarkR[j] then

for i:=1 to z do

if (a[i,j] = 0)and (isMark(i,j)) then

if MarkZ[i] then

begin

MarkZ[i]:=false;

sw:=true;

end;

end;

//Ищем значение минимального элемента, из матрицы, для которой соответствующее

//значение MarkR и MarkZ равно "false".

min:=2147483647;

for i:=1 to z do

if not(MarkZ[i]) then

for j:=1 to r do

if not(MarkR[j]) then

if (a[i,j]<min)and(a[i,j]>0) then

min:=a[i,j];

for i:=1 to z do

if not(MarkZ[i]) then

for j:=1 to r do

a[i,j]:=a[i,j]-min;

for j:=1 to r do

if MarkR[j] then

for i:=1 to z do

a[i,j]:=a[i,j]+min;

end;

begin

InPut;

Step1;

Step2;

Step3;

Step4;

Step3;

write('Done!');

readln;

end.

// \*Функции сортировки вектора примечательны тем, что результатом их выполнения

// есть вектор, в котором лежат индексы элементов исходного (ещё не отсортиро-

// ваного) вектора в порядке убывания соответствующих элементов, например:

// Исходнаый вектор: 2, 1, 1, 1, 4, 1

// Результат: 5, 1, 2, 3, 4, 6

1. **Результати роботи програми**

Вхідний файл:

6 6

8 5 -1 7 6 9

6 3 9 1 -1 7

5 -1 3 6 8 8

9 1 8 -1 5 4

1 2 1 -1 2 3

-1 -1 3 5 3 10

Результат:

..:: Iteration 1 ::..

i=6 j=5

i=4 j=2

i=3 j=3

i=2 j=4

i=5 j=6

..:: Iteration 2 ::..

i=3 j=3

i=2 j=4

i=6 j=5

i=4 j=6

i=1 j=2

i=5 j=1

Done!