Міністерство освіти і науки України

Національний технічний університет України

«Київський Політехнічний Інститут»

Факультет прикладної математики

Кафедра Спеціалізованих комп’ютерних систем

Лабораторна робота №4

З дисципліни «Системне програмне забезпечення» :

«Дослідження принципів проектування та роботи динамічного та статичного планування»

Варіант №16

Виконав:

студент групи КВ-92

Степанюк М.Ф.

Перевірив:

Симоненко В. П.

Київ 2012

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

**Алгоритм:**

Твердження 1

Оскільки кожен рядок МС є віддзеркалення можливого розподілу заявок на ресурси, то зміна місцями двох рядків МС із запам'ятовуванням нового розташування заявок не впливає на якісну характеристику МС і початкового графа.

Твердження 2

Оскільки кожен стовпець МС є віддзеркалення претендування заявок на захоплення даного ресурсу, то зміна місцями двох стовпців МС із запам'ятовуванням нового розташування ресурсів не впливає на якісну характеристику МС і початкового графа.

Теорема 3

Якщо після еквівалентного перетворення МС вдається отримати терм-ранг МС рівний рангу МС, то всі заявки мають місця призначення, відповідні "1" головної діагоналі і: знайдене повне паросочетаніє, отриманий один з варіантів рішення задачі призначення заявок по ресурсах

Teoрема 5.

Якщо в матриці RJ[i,j], i=1..N, j=1..N, можна виділити підматрицю MN[k,l], k=1..T, l=(N-S+1)..N, де S+T=N і MN=θ, тоді для Ri∈{1,..,(N-S)}, Jj∈{(T+1),..,N}: ∀(Ri ,Jj)∉A. Все (Ri ,Jj)∉A

Teoрема 6

Якщо в матриці MT[i,j], i=1..N, j=1..N, можна виділити декілька підматриць, що задовольняють Теоремі 5, то все відповідні симетричні їм, щодо головної діагоналі, підматриці є "конфліктними" і повинні бути обнулені (мал. 3.а-д).

Згідно зтвердженнями 1-3 виконуємо наступні кроки:

1. Стовпці, які містять найбільшу кількість одиниць переміщуємо в ліву частину матриці
2. Рядки, які містять найбільшу кількість нулів переміщуємо в верхню частину матриці
3. Після даних перетворень отримуємо «1», які розміщені на головній діагоналі. Якщо в матриці MС[i,j], i=1..N, j=1..N можна виділити підматрицю MM[k,l], k=1..T, l=(N-S+1)..N, где S+T>N и MM=θ (θ — нульова матриця), тоді завдання призначення не має повного рішення
4. Згідно за теоремою 5 та 6 обнуляємо під матрицю, симетричну нулевій під матриці відносно головної діагоналі як конфліктну зону.

***Лістинг програми:***

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SPZ\_4

{

class matrix

{

static int size = 6;

int[,] matr = new int[size, size];

int[,] start\_matr = new int[2, size];

int subm=0, subn=0;

int[] vect = { 1,0,1,0,1,0,

0,1,1,1,1,1,

1,0,1,0,1,0,

0,1,1,1,1,1,

1,0,1,0,1,0,

1,0,1,0,1,0};

int[] vect2 = {

1,1,1,0,0,0,

1,1,1,1,0,1,

1,1,1,0,0,0,

1,1,1,1,1,0,

1,1,1,1,1,1,

1,1,1,1,1,1

};

public matrix()

{

Random rnd = new Random();

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

{

//matr[i, j] = rnd.Next(0, 2);

matr[i, j] = vect[i\*size + j];

}

for (int i = 0; i < size; i++)

{

start\_matr[0, i] = i;

start\_matr[1, i] = i;

}

}

public void PrintMatr()

{

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

Console.Write("{0} ", matr[i, j]);

Console.Write("\n");

}

}

int FndMaxCol( int start\_pos )

{

int max\_col = -1, max\_num = 0;

int num = 0;

for (int j = start\_pos; j < size; j++)

{

num = 0;

for (int i = 0; i < size; i++)

if (matr[i, j] > 0)

num++;

if (num > max\_num)

{

max\_col = j;

max\_num = num;

}

}

return max\_col;

}

int FndMaxRow(int start\_pos)

{

int max\_row = -1, max\_num = 0;

int num = 0;

for (int i = start\_pos; i < size; i++)

{

num = 0;

for (int j = 0; j < size; j++)

if (matr[i, j] == 0 )

num++;

if (num > max\_num)

{

max\_row = i;

max\_num = num;

}

}

return max\_row;

}

void ExchangeCol(int m, int n)

{

int pos = start\_matr[1, m];

start\_matr[1, m] = start\_matr[1, n];

start\_matr[1, n] = pos;

for (int i = 0; i < size; i++)

{

int buf = matr[i, n];

matr[i, n] = matr[i, m];

matr[i, m] = buf;

}

}

void ExchangeRow(int m, int n)

{

int pos = start\_matr[0, m];

start\_matr[0, m] = start\_matr[0, n];

start\_matr[0, n] = pos;

for (int i = 0; i < size; i++)

{

int buf = matr[n, i];

matr[n, i] = matr[m, i];

matr[m, i] = buf;

}

}

public void Transform()

{

for (int i = 0; i < size; i++)

{

int col = FndMaxCol(i);

if (col < 0 || i == col )

continue;

ExchangeCol(i, col);

}

for (int i = 0; i < size; i++)

{

int row = FndMaxRow(i);

if (row < 0 || i == row )

continue;

ExchangeRow(i, row);

}

}

bool CheckZeroMatr(int m,int n)

{

for ( int i = 0; i<m; i++)

for (int j = n; j < size; j++)

if (matr[i, j] == 1 )

return false;

return true;

}

public void FndSubMatrix()

{

int m = size, n = 0;

for ( int i = size-2; i>0; i-- )

for (int j = i-1; j < size; j++)

{

if (CheckZeroMatr(i, j))

{

if ((i \* (size-j)) > (subm \* (size-subn)))

{

subm = i;

subn = j;

}

}

}

}

public void FndSolution()

{

Console.WriteLine("SubMatrix starts from {0} col and {1} row", subn, subm);

for (int i = 0; i < subm; i++)

for (int j = subn; j < size; j++)

{

if ( i != j )

matr[j,i] = 0;

}

PrintMatr();

}

public void ShowMatrix()

{

matrix buf = new matrix();

for (int i = 0; i < size; i++)

{

int ii = start\_matr[0, i];

for (int j = 0; j < size; j++)

{

int jj = start\_matr[1, j];

buf.matr[ii, jj] = matr[i, j];

}

}

buf.PrintMatr();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SPZ\_4

{

class Program

{

static void Main(string[] args)

{

matrix m = new matrix();

m.PrintMatr();

//Console.WriteLine();

//m.ShowMatrix();

Console.WriteLine();

m.Transform();

Console.WriteLine();

m.PrintMatr();

m.FndSubMatrix();

m.FndSolution();

Console.WriteLine();

m.ShowMatrix();

}

}

}