Национальный технически университет Украины «Киевский политехнический институт»

Факультет информатики и вычислительной техники

Кафедра вычислительной техники

Лабораторная работа №4

По дисциплине «Системное программное обеспечение»

Выполнил: студент 3-го курса,

гр. ИО-91Щербина Александр

Киев -2012

№ЗК=9118

Вариант = 18mod13+1=6

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

**Описание алгоритма**

1. Поиск строки матрицы, где сумма элементов минимальна.
2. Поменять местами первую строку матрицы с той что найдена в п1.
3. Поиск столбца с минимально сумой элементов среди тех, у которых первый элемент равен 1.
4. Поменять местами первый столбец матрицы с тем что найден в п3.
5. Заменить все элементы первого столбца и первой строки 0 кроме первого элемента, который равен 1.
6. Заявку номер которой равен номеру столбца, отправить на ресурс номер которого равен номеру строки.
7. Повторить операции 1-6 для подматрицы, которую получаем из матрицы п5 исключение первого столбца и строки, но выполнять операции обмена между столбцами и строками для самой начальной матрицы.

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

**public** **class** Planner {

**public** **static** **void** main(String[] args){

**int**[][] matrix=**new** **int**[][]{{0,1,0,1,0},{1,0,1,1,1},{0,0,0,0,1},{1,1,0,0,0},{0,1,1,1,0}};

matrix=*plann*(matrix);

}

**public** **static** **int**[][] plann(**int**[][] ms){

**int**[] xName=**new** **int**[ms[0].length];

**int**[] yName=**new** **int**[ms.length];

**for**(**int** i=0;i<xName.length;i++){

xName[i]=i;

}

**for**(**int** j=0;j<yName.length;j++){

yName[j]=j;

}

System.*out*.println("Start matrix");

*showMatrix*(ms, xName, yName);

**int** minSize=ms.length;

**if**(minSize>ms[0].length){

minSize=ms[0].length;

}

**for**(**int** i=0;i<minSize;i++){

**int** row=*getMinRow*(ms, i, i);

ms=*xchangeRows*(ms, i, row);

*xchange*(yName, i, row);

**int** col=*getMinColumn*(ms, i, i);

ms=*xchangeColumns*(ms, i, col);

*xchange*(xName, i, col);

ms=*setZeros*(ms, i);

}

**for**(**int** i=0;i<xName.length-1;i++){

**int** p=*getIndex*(xName, i);

**if**(p==-1){

**throw** **new** IllegalArgumentException();

}

ms=*xchangeColumns*(ms, i, p);

*xchange*(xName, i, p);

}

**for**(**int** i=0;i<yName.length-1;i++){

**int** p=*getIndex*(yName, i);

**if**(p==-1){

**throw** **new** IllegalArgumentException();

}

ms=*xchangeRows*(ms, i, p);

*xchange*(yName, i, p);

}

System.*out*.println("Result matrix");

*showMatrix*(ms, xName, yName);

**return** ms;

}

**public** **static** **int** getIndex(**int**[] names,**int** n){

**for**(**int** i=0;i<names.length;i++){

**if**(n==names[i]){

**return** i;

}

}

**return** -1;

}

**public** **static** **void** showMatrix(**int**[][]ms,**int**[] xName,**int**[] yName){

System.*out*.print(" ");

**for**(**int** i=0;i<xName.length;i++){

System.*out*.print(xName[i]+" ");

}

System.*out*.println();

**for**(**int** i=0;i<ms.length;i++){

System.*out*.print(yName[i]+" ");

**for**(**int** j=0;j<xName.length;j++){

System.*out*.print(ms[i][j]+" ");

}

System.*out*.println();

}

}

**public** **static** **int**[][] setZeros(**int**[][] ms,**int** onePosition){

**for**(**int** i=onePosition+1;i<ms.length;i++){

ms[i][onePosition]=0;

}

**for**(**int** i=onePosition+1;i<ms[0].length;i++){

ms[onePosition][i]=0;

}

**return** ms;

}

**public** **static** **int**[] xchange(**int**[] array,**int** p1,**int** p2){

**int** buf=array[p1];

array[p1]=array[p2];

array[p2]=buf;

**return** array;

}

**public** **static** **int** getMinRow(**int**[][] ms,**int** stI,**int** stJ){

**int** min=Integer.*MAX\_VALUE*;

**int** n=-1;

**for**(**int** i=stI;i<ms.length;i++){

**int** sum=0;

**for**(**int** j=stJ;j<ms[0].length;j++){

sum+=ms[i][j];

}

**if**(sum<min){

min=sum;

n=i;

}

}

**return** n;

}

**public** **static** **int** getMinColumn(**int**[][] ms,**int** stI,**int** stJ){

**int** min=Integer.*MAX\_VALUE*;

**int** n=-1;

**for**(**int** j=stJ;j<ms[0].length;j++){

**if**(ms[stI][j]==0){

**continue**;

}

**int** sum=0;

**for**(**int** i=stI;i<ms.length;i++){

sum+=ms[i][j];

}

**if**(sum<min){

min=sum;

n=j;

}

}

**return** n;

}

**public** **static** **int**[][] xchangeColumns(**int**[][] ms,**int** col1,**int** col2){

**for**(**int** i=0;i<ms.length;i++){

**int** buf=ms[i][col1];

ms[i][col1]=ms[i][col2];

ms[i][col2]=buf;

}

**return** ms;

}

**public** **static** **int**[][] xchangeRows(**int**[][] ms,**int** row1,**int** row2){

**for**(**int** i=0;i<ms[0].length;i++){

**int** buf=ms[row1][i];

ms[row1][i]=ms[row2][i];

ms[row2][i]=buf;

}

**return** ms;

}

}

Пример работы программы

Start matrix

0 1 2 3 4

0 0 1 0 1 0

1 1 0 1 1 1

2 0 0 0 0 1

3 1 1 0 0 0

4 0 1 1 1 0

Result matrix

0 1 2 3 4

0 0 1 0 0 0

1 0 0 0 1 0

2 0 0 0 0 1

3 1 0 0 0 0

4 0 0 1 0 0