**算法训练 关联矩阵**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　有一个n个结点m条边的有向图，请输出他的关联矩阵。

输入格式

　　第一行两个整数n、m，表示图中结点和边的数目。n<=100,m<=1000。  
　　接下来m行，每行两个整数a、b，表示图中有(a,b)边。  
　　注意图中可能含有重边，但不会有自环。

输出格式

　　输出该图的关联矩阵，注意请勿改变边和结点的顺序。

样例输入

5 9  
1 2  
3 1  
1 5  
2 5  
2 3  
2 3  
3 2  
4 3  
5 4

样例输出

1 -1 1 0 0 0 0 0 0  
-1 0 0 1 1 1 -1 0 0  
0 1 0 0 -1 -1 1 -1 0  
0 0 0 0 0 0 0 1 -1  
0 0 -1 -1 0 0 0 0 1

本题的C++参考代码如下：

#include<cstdio>

struct group

{

int first;

int second;

};

int main()

{

int n,m,i,j;

scanf("%d%d",&n,&m);

struct group g[m];

for(i=1;i<=m;i++)

{

scanf("%d%d",&g[i].first,&g[i].second);

}

// for(i=1;i<=m;i++)

// {

// printf("(%d %d)\n",g[i].first,g[i].second);

// }

for(i=1;i<=n;i++)

{

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

{

if(g[j].first==i)

printf("1 ");

else if(g[j].second==i)

printf("-1 ");

else

printf("0 ");

}

printf("\n");

}

return 0;

}

本题的C参考代码如下：

#include<stdio.h>

int main()

{

int i, ii,n,m, a[1000][2];

scanf("%d%d", &n, &m);

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

scanf("%d%d", &a[i][0], &a[i][1]);

for ( i = 1; i <=n; i++)

{

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

{

if (i==a[ii][0])

{

printf("1 ");

}

else

if (i==a[ii][1])

{

printf("-1 ");

}

else

{

printf("0 ");

}

}

printf("\n");

}

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class Main {

public void method1() throws NumberFormatException, IOException{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

String first = br.readLine();

String[] nm = first.split(" ");

int n = Integer.parseInt(nm[0].trim());

int m = Integer.parseInt(nm[1].trim());

int a[][] = new int[n][m];

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

String second = br.readLine();

String[] pq = second.split(" ");

int p = Integer.parseInt(pq[0].trim());

int q = Integer.parseInt(pq[1].trim());

a[p-1][i] = 1;

a[q-1][i] = -1;

}//end for

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

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

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

}

if (i!=n-1) {

System.out.println();

}

}

}

public static void main(String[] argStrings) throws IOException {

Main main = new Main();

main.method1();

}

}