

## 实验 6 图的应用——通信网络

### 一、问题分析

#### 1. 分析并确定要处理的对象（数据）是什么

问题求解是找出  $N$  个部门之间互相了解所有部门的个数。条件已知两个部门之间的单项联系。由于各个部门之间具有多对多的关系，可以确定需要一种非线性的结构去存储部门之间的关系，简化后是一个有  $n$  个顶点， $m$  条边的有向图的通路问题。题目要求我们求的是有多少各部门知道所有部门的存在，即判断每个点与其他所有点之间是否连通，划归为有向图的连通问题的求解。

#### 2. 分析并确定要实现的功能是什么

实现的功能：求出每个点与之连通的点的个数，进而判断具体问题中有多少各部门知道所有部门的存在。

#### 3. 分析并确定处理后的结果如何显示

如何去求解图中每个点的连通性？可以通过 **warshall** 算法去求出任意两个点之间是否连通，用 **warshall** 算法得到的矩阵  $M$ ，由于题目中要求包含本部门在内，所以在矩阵  $M$  中对角线处每一个位置都要赋值为 1，表示包括本部门在内。之后判断是否某个部门知道有  $N$  个部门，只需要看每一对对称位置是否全为 0，若全为 0，则说明该部门不可能知道所有部门的存在，从反面着手来处理可以在一定程度上减少计算量。遍历矩阵中每一个点，用一个变量 **flag** 去进行刚刚的判断。

### 二、数据结构和算法设计

#### 1. 抽象数据类型设计

考虑到用 **warshall** 算法来看图的连通性，可以设计用邻接矩阵来实现图的存储，矩阵中每个点的坐标表示各点的序号，矩阵中点的值表示有无边相连。**ADT** 中还需要有一个标记数组来标记顶点，一个顶点数组去存储顶点信息，一个 **bool** 型变量去判断图的类型，两个变量 **numVert**, **numEdge** 分别表示顶点个数和图之间的边数。

基本操作需要实现对图的初始化，返回顶点个数，边数，在存储图的时候需要在邻接矩阵中设置顶点信息和边权，用 **setVex**, **setEdge** 两个函数来实现。**Warshall** 算法用一个函数对邻接矩阵进行操作，求解问题通过遍历整个图来进行统计  $N$  个点之间互通的个数，这里也设计一个函数来实现。

#### 2. 物理数据对象设计（不用给出基本操作的实现）

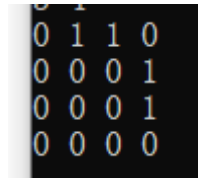
##### 1) 算法思想的设计

利用 **warshall** 算法可以求出任意两个点之间是否连通，**warshall** 算法的主要思想就是遍历邻接矩阵，找到每  $j$  列中为 1 的位置  $(i,j)$ ，第  $i$  行就等于第  $i$  行和第  $j$  行析取的结果，遍历每一行就可以得到一个表示出任意两点之间是否连通的矩阵。原理就是，找到第  $j$  列中为 1 的位置  $(i,j)$  说明  $i$  和  $j$  连通，然后再看  $j$  和那些点相连通，即去看第  $j$  行中为 1 的位置，故将两行析取就可以得到和  $i$  相连通的点。遍历矩阵中所有为 1 的位置就可以得到整个图的连通关系。本质上就是对矩阵中元素进行析取运算。

用 **warshall** 算法得到了一个图的传递闭包，要满足题意，需要遍历传递闭包中的每一行或者每一列都全为 1，全为 1 说明此点和其余所有点都连通，从

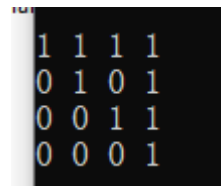
正面求解太繁杂，可以看每一行点和其对称位置处的点是否全为 0，全为 0 说明不可能满足和其余所有点都连通的条件，则从下一个点继续判断。注意自身连通需要手动补上，因为有的邻接矩阵中可能得不到自身连通。

2) 请用题目中样例，基于所设计的算法，详细给出样例求解过程。



0	1	1	0
0	0	0	1
0	0	0	1
0	0	0	0

题目中给的样例转化为邻接矩阵，遍历找到第二列为 1 位置，将第 2 行和第 1 行析取，再次遍历，将第 3 行和第 1 行析取，最后将第 2 行，第 3 行分别和第 4 行析取，得到图的传递闭包，手动将邻接矩阵中对角线上的



1	1	1	1
0	1	0	1
0	0	1	1
0	0	0	1

0 赋值为 1，得到所示矩阵，观察矩阵发现第 1 行和第 4 列全为 1，故有两个部门可以知道其他 4 个部门的存在。

### 3. 关键功能的算法步骤

**Warshall 算法的算法步骤：**遍历邻接矩阵的每一列  $j$ ，找到每  $j$  列中为 1 的位置  $(i,j)$ ，第  $i$  行就等于第  $i$  行和第  $j$  行析取的结果，重复以上步骤直到遍历完成整个邻接矩阵，即可得到图的传递闭包。

**判断某个点是否和其余所有点连通的算法步骤：**首先将传递闭包中对角线位置全赋值为 1，遍历传递闭包中的每一行中的点，看每一行点和其对称位置处的点是否全为 0，全为 0 说明不可能满足和其余所有点都连通的条件，则从下一个点继续判断。这里用一个 `bool` 变量，处置设为 1，当遇到不满足是将其设置为 0，每一行遍历完后，进行统计满足条件的点数。

## 三、算法性能分析

**算法时间复杂度的分析：**首先看将输入的点与点之间的关系存储邻接矩阵部分，设置顶点信息用到 `for` 循环，循环  $n$  次，时间复杂度为  $O(n)$  设置边权同样也用到 `for` 循环，循环  $n$  次，时间复杂度为  $O(n)$ ，warshall 算法中利用到了二重循环去遍历邻接矩阵，找到为 1 的位置，需要遍历每一行进行析取运算，循环  $n$  次，时间复杂度为  $O(n^3)$ ，求解某点与其余所有点连通点的个数，用到二重循环遍历每一个点进行判断，时间复杂度为  $O(n^2)$ 。综合以上，根据化简规则，总的时间复杂度为  $O(n^3)$ 。

**空间复杂度的分析：**初始化邻接矩阵，动态开辟一个大小为  $N$  的标记数组，其余操作均无动态内存创建，故空间复杂度为  $O(n)$ 。