中山大学数据科学与计算机学院本科生实验报告

课程名称：编译器构造实验 任课教师：陈炬桦 教学助理（TA）：

|  |  |  |  |
| --- | --- | --- | --- |
| 学年度 | 2018-2019 | 学期 | 第二学期 |
| 年级 | 2016级 | 专业（方向） | 计算机科学与技术 |
| 学号 | 16337233 | 姓名 | 王凯祺 |
| 电话 | 13434386182 | Email | [Wangkq3@mail2.sysu.edu.cn](mailto:Wangkq3@mail2.sysu.edu.cn) |
| 开始日期 | 2019/4/26 | 完成日期 | 2019/4/26 |

1. 实验题目

**传递闭包的Warshall算法**

**1.1 Description**

传递闭包 的Warshall算法为：  
for j:=1 until N do  
for i:=1 until N do  
 if M[i,j]=1 then   for k:=1 until N do  
        M[i,k]:= M[i,k]∨M[j,k]  
输入矩阵M(R),计算并输出M(R+)。

**1.2 Input**

输入M方阵的行列数；  
输入矩阵M(R);

**1.3 Output**

计算并输出M(R+)，每个符号占3格；

2. 算法描述(介绍程序模块功能；流程图)

算法为三重for循环。我们可以把状态写成三维（f[k][i][j]），f[k][i][j]表示的是只经过1…k这些节点的情况下，i到j是否可达。考虑从k转移到k+1，若i🡪j不需要经过k+1，则f[k+1][i][j]就是f[k][i][j]；若i🡪j需要经过k+1，则k+1只会经过最多一次（若经过两次，必有k+1到k+1的路径，可以删去），我们只需要检查是否能从i走到k+1，以及从k+1走到j。

在实际操作中，我们可以把第一维省略。

3. 测试数据

样例输入

5

0 1 1 0 0

1 0 1 0 1

0 1 0 0 1

0 0 0 0 0

1 0 0 0 1

样例输出

1 1 1 0 1

1 1 1 0 1

1 1 1 0 1

0 0 0 0 0

1 1 1 0 1

4. 程序清单

#include <bits/stdc++.h>

#define N 10

**using** **namespace** std;

**int** n, g[N][N];

**int** main() {

scanf("%d", &n);

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

**for** (**int** j = 1; j <= n; ++j)

scanf("%d", &g[i][j]);

**for** (**int** k = 1; k <= n; ++k)

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

**for** (**int** j = 1; j <= n; ++j)

**if** (g[i][k] && g[k][j])

g[i][j] = 1;

**for** (**int** i = 1; i <= n; ++i) {

**for** (**int** j = 1; j <= n; ++j)

printf("%3d", g[i][j]);

puts("");

}

}