离散数学课程设计

项目说明文档

Warshall算法求传递闭包

|  |  |
| --- | --- |
| 作者姓名： | 杨烜赫 |
| 学 号： | 2252709 |
| 指导教师： | 李 冰 |
| 学院专业： | 软件学院 软件工程 |



同济大学

Tongji University

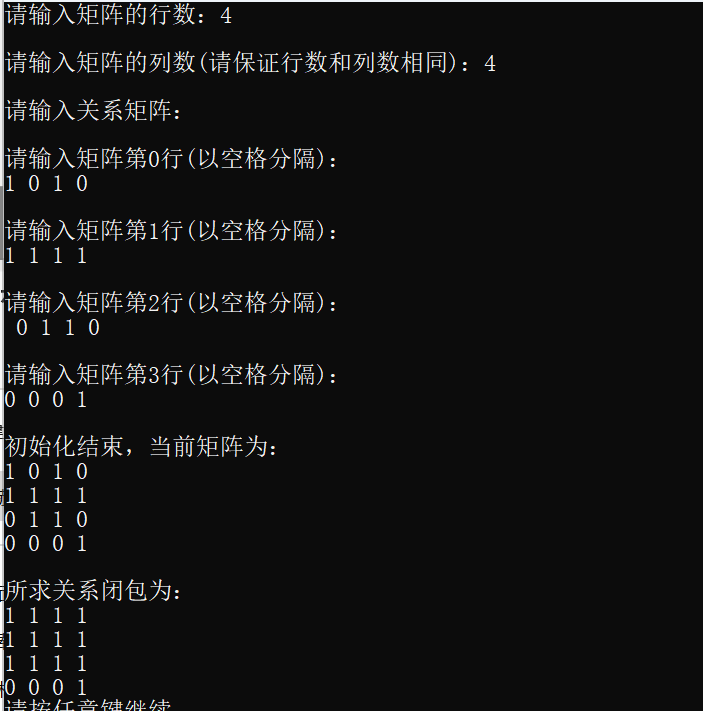
1. **题目简介**
2. **实验目的**

根据给定的关系矩阵，利用 Warshall 算法高效求解传递闭包。

**2.实验内容**

实现Warshall 算法高效求解传递闭包。

**3.项目示例**

****

1. **项目实现**

## 实验原理

离散数学定义：

t(R) = R u R^2 u R^3 u..... 其中R^(n+1) = R^n 复合 R

矩阵表示：

M（R） = M + M^2 + M^3 +....+M^n(其中加为逻辑加）

所以我们只要按照这个公式每次更新M，最后的Mn就是传递闭包

Warshall算法：

(1)置新矩阵A＝M；

(2)i＝1；

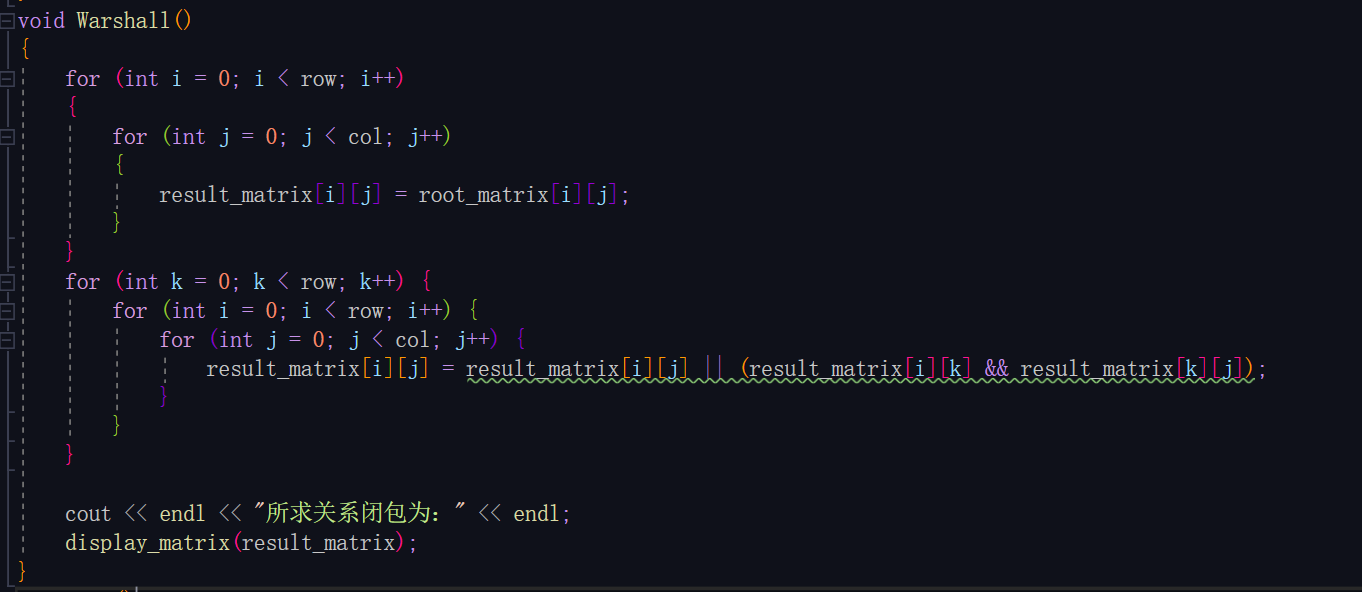
(3)对所有j如果A[j，i]＝1，则对k＝1，2，…，n，A[j，k]＝A[j，k]∨A[i，k]；

(4)i加1；（i是行，j是列）

(5)如果i≤n，则转到步骤3)，否则停止。

思想：不难理解,对于每个相通的j - > i,我们可以从这个相通关系出发，看看能不能通过这条相通的j - > i,更新一下j - >k。对所有的可通关系都更新一遍M，最后的结果就是传递闭包了！

## 2.实现过程



**Warshall 算法实现**

**初始化：**

首先，将原始矩阵 root\_matrix 的内容复制到结果矩阵 result\_matrix 中。这里，root\_matrix 表示图的邻接矩阵，其中 root\_matrix[i][j] 表示顶点 i 到顶点 j 是否有直接的路径。

**计算传递闭包：**

使用三层嵌套循环来迭代更新 result\_matrix，以计算传递闭包。外层循环变量 k 代表中间顶点。对于每个中间顶点 k，算法检查通过 k 是否可以找到从 i 到 j 的新路径。如果 result\_matrix[i][k] && result\_matrix[k][j] 为 true，意味着存在从 i 经过 k 到 j 的路径，因此将 result\_matrix[i][j] 设置为 true。

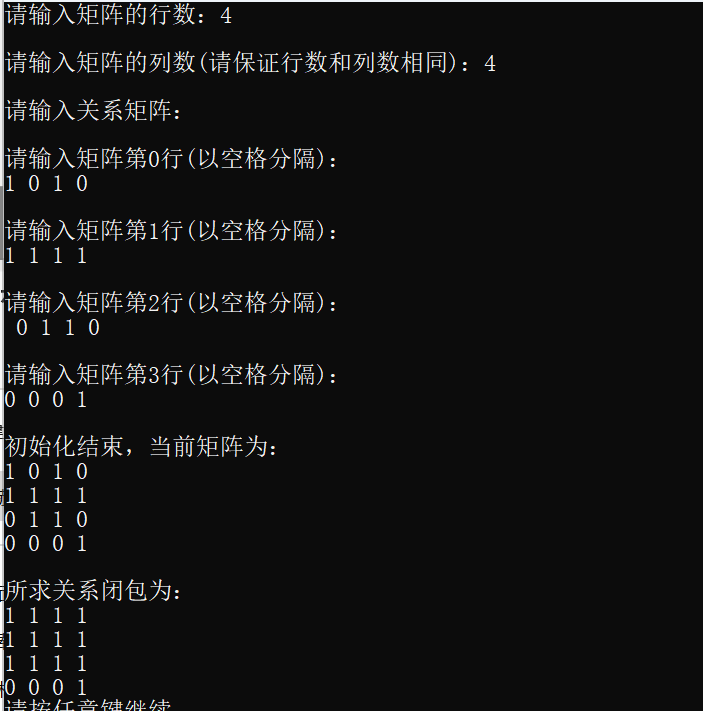
这个步骤确保了考虑所有可能的通过中间顶点的路径。

**输出结果：**

在完成传递闭包的计算后，通过调用 display\_matrix 方法在控制台输出最终的传递闭包矩阵。

**三、项目测试**

结果测试：

****

**实际值与理论值相同，与项目三结果也一致**

**健壮性测试在项目三已经完成**

**四、心得与总结**

在本次实验中，通过运用 Warshall 算法，我成功规避了进行矩阵合成运算这一高时间复杂度的步骤，有效地降低了求传递闭包的时间成本。这一过程再次强调了在算法设计中对时间复杂度的重要考虑。在之前的学习中，我曾经接触过闭包的概念，并通过矩阵快速幂方法来高效地实现矩阵运算。这种方法在当时给我留下了深刻的印象，因为它展示了如何通过优化算法来提高计算效率。

然而，我遇到了 Warshall 算法，它从一个全新的视角出发，提供了另一种根据关系矩阵求传递闭包的有效方法。这种方法不仅令人耳目一新，还激发了我对不同算法设计思路的深入思考。

总体而言，这次实验不仅加深了我对传递闭包概念的理解，还拓展了我在算法设计上的视野。从矩阵快速幂到 Warshall 算法，每种方法都有其独特的应用场景和优势，这使我意识到在解决问题时灵活选择合适的算法是至关重要的。这种认识无疑将对我未来在计算机科学领域的学习和实践产生深远的影响。