

北京邮电大学

实验报告



实验名称: 使用队列和栈判断回文串

班级 : 2019211309 姓名 : 陈悦 学号 : 2019211413 分工 : 文档

班级 : 2019211309 姓名 : 马晓亮 学号 : 2019211400 分工 : 代码

2020 年 10 月 21 日

一 需求分析

本程序可以完成两个矩阵相乘或相减的计算任务。输入一个矩阵的格式为

$m\ n\ p$

$i_1\ j_1\ val_1$

$i_2\ j_2\ val_2$

$i_3\ j_3\ val_3$

.....

$i_p\ j_p\ val_p$

m 表示矩阵的行数, n 表示矩阵的列数, p 表示稀疏矩阵中结点的个数。

本程序要求用户输入两个矩阵 A、B, 然后再选择计算方式。然后程序会在后端进行计算, 将计算结果保存在矩阵 C 中。最后程序将 C 输出到终端, 输出格式同样使用三元组进行输出, 顺序为先行后列。

同时, 我们对于用户的输入作出如下限制:

1. $n > 0$ 且 $m > 0$ 且 $p \geq 0$
2. 对于一个矩阵, 数值的下标应该保持在矩阵范围内 (从 1 开始), 不得超出矩阵。
3. 必须要输入两个矩阵后再进行运算。

对于不符合输入限制要求的操作, 程序将在终端上输出 (Input Error)

同时, 用户应该对输入的矩阵是否能进行运算进行检查。如果输入的矩阵不能进行运算, 程序将返回 (Math Error). 可能的原因分别有:

1. Error Code 0: 相加的两个矩阵的行数与列数不相等。
2. Error Code 1: 两个矩阵不能相乘

二 概要设计

程序由三个模块组成, 分别为 *Matrix*, *Solution* 和 *main*

Matrix 是一种数据类, 定义了稀疏矩阵类以及它的运算方法, 包括在矩阵中插入元素。

Solution 是可以与用户交互的类, 定义了如何输入和输出, 以及调用 *Matrix* 中的方法完成计算。

main 为程序的主函数，主要用于初始化 *Solution* 和调用 *Solution* 中与用户交互的方法。主程序的流程如下：

1. 初始化 *Solution*
2. 调用输入方法
3. 调用计算方法
4. 调用输出方法
5. 释放内存

三 详细设计

四 调试分析报告

五 用户使用说明

六 测试结果