**要求代码和实验报告规范，在算法思想中：对实验涉及的数据结构进行有效设计和分析；对算法进行分析并给出时间、空间复杂度的结论；清晰表达实验思路、出现的问题及解决方法。**

一、调试成功程序及说明

1、

题目：输入稀疏矩阵，建立稀疏矩阵三元组顺序结构，实现矩阵的列序遍历转置和快速转置算法。

算法思想：

1. 算法思想与设计思路

列序遍历法，就是对原三元组进行col次遍历，每次找出相同列的元素逆置。

快速转置法是采用额外两个数组，分别存储每列的非零元素的个数和该列中非零元素在三元组中的位置。

1. 算法复杂度分析

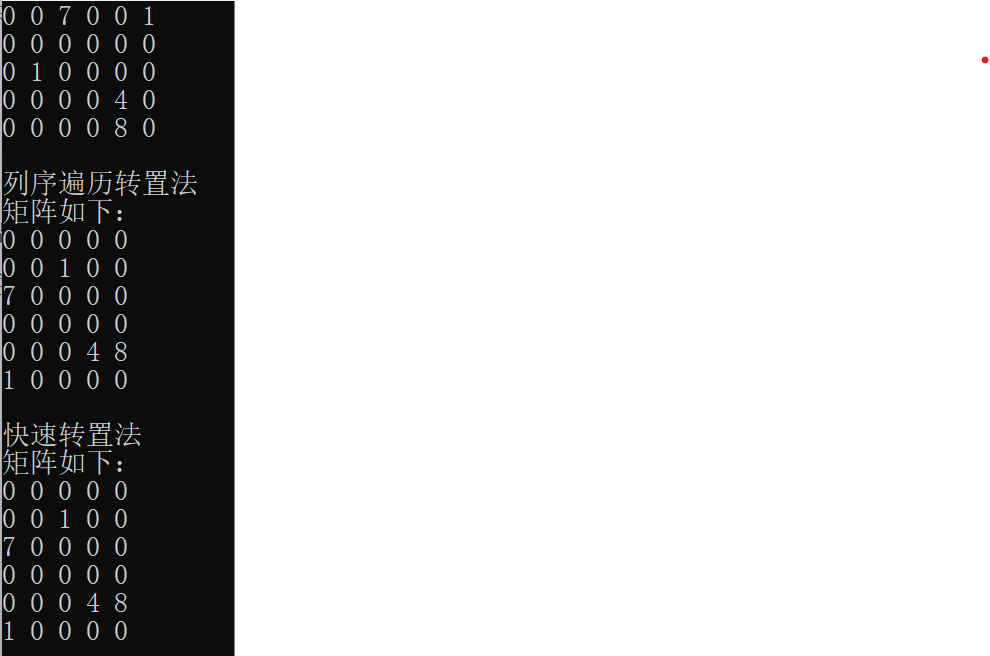
列序遍历法的时间复杂度：O（col\*num）

快速转置法的时间复杂度：O（col+num）

1. 实现过程中遇到的问题与解决方法

没有问题

运行结果：



结果分析：与预期相同

2、

题目：求矩阵的马鞍点。（书P69 7）

算法思想：

1. 算法思想与设计思路

从头往尾遍历，找到每一行的行最小值和每一列的列最大值并将它们存储起来，然后遍历，如果列最大值等于行最小值就是马鞍点

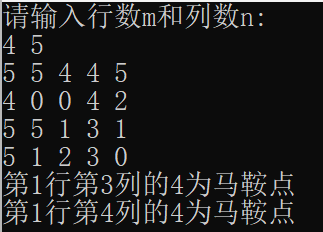
1. 算法复杂度分析

时间复杂度为O(M\*N),空间复杂度为O(M+N)

1. 实现过程中遇到的问题与解决方法

没有问题

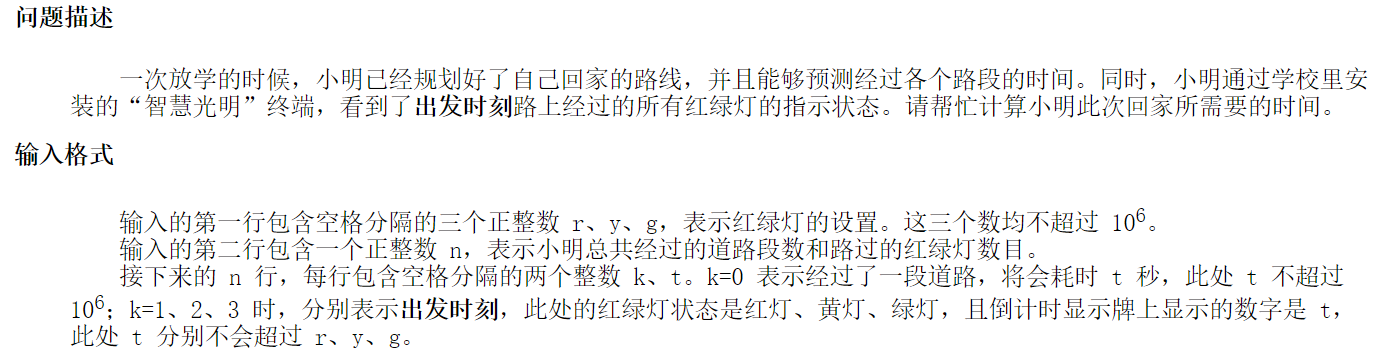
运行结果：



结果分析：与预期相同

3.

题目：



算法思想：

1. 算法思想与设计思路

sum代表一次交通变化灯的循环时间。用红绿灯在出发时刻走过的时间+小明到达该此处的所需时间的总值对sum进行取余，对余数v进行判断，算出等待时间。

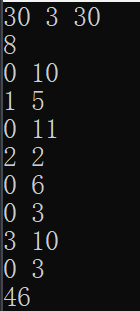
1. 算法复杂度分析

时间复杂度为O(N)，空间复杂度为O（1）

1. 实现过程中遇到的问题与解决方法

没有问题

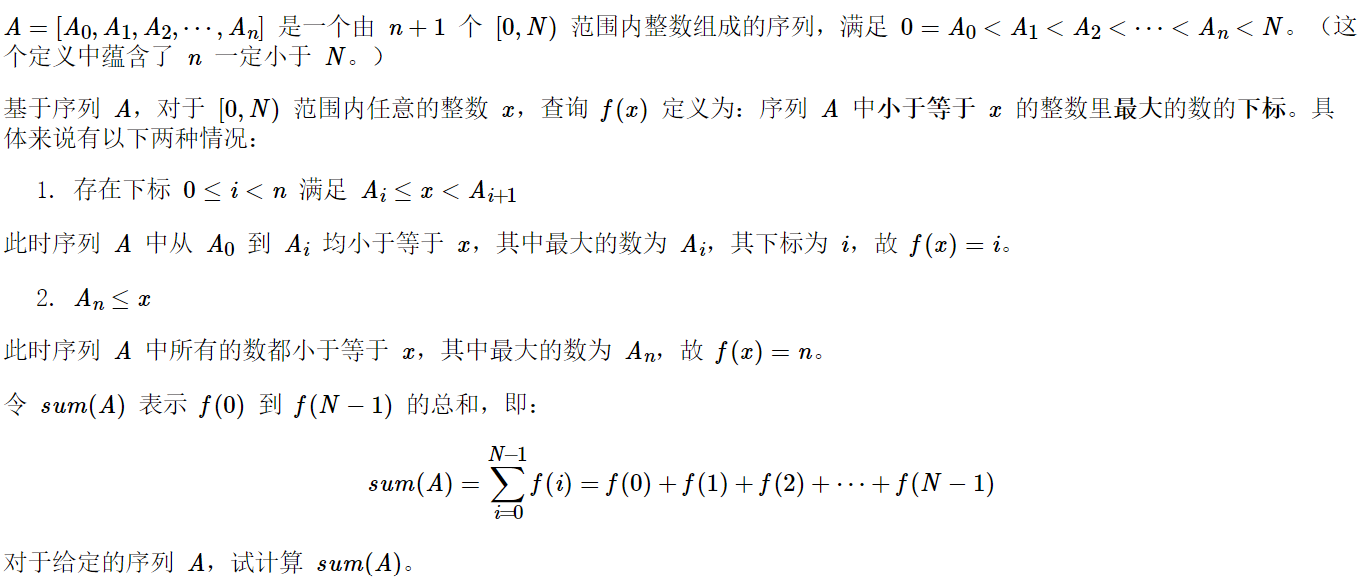
运行结果：



结果分析：与预期相同

4.

题目：



算法思想：

1. 算法思想与设计思路

依次输入数组元素，对[Ai,Ai+1)范围内的数，用i\*（Ai+1-Ai）计算，最后计算N\*(N-An)

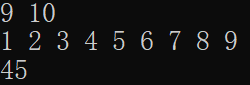
1. 算法复杂度分析

时间复杂度为O(N),空间复杂度为O(1)

1. 实现过程中遇到的问题与解决方法

没有问题

运行结果：



结果分析：与预期相同

附加题：

3.20题目：假设以二维数组 g(1…m, 1…n)表示一个图像区域，g[i,j]表示该区域中点(i,j)所具颜色，其值为从 0 到 k 的整数。编写算法置换点(i 0 ,j 0 )所在区域的颜色。约定和(i 0 ,j 0 )同色的上、下、左、右的邻接点为同色区域的点。

算法思想：

1. 算法思想与设计思路

记录原有颜色，向四个方向查找，符合条件则加入栈中，弹栈时修改颜色

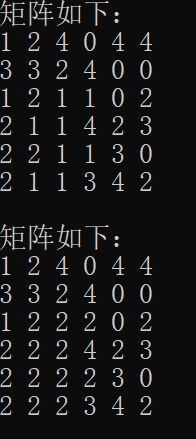
1. 算法复杂度分析

M为同区域点个数，时间复杂度为O(M),空间复杂度为O(M)

1. 实现过程中遇到的问题与解决方法

没有问题

运行结果：



结果分析：与预期相同

3.28

题目：假设以带头结点的循环链表表示队列，并且只设一个指针指向队尾元素结点（注意不设头指针），试编写相应的队列初始化、入队列和出队列的算法。

算法思想：

1. 算法思想与设计思路

初始化尾指针为空，入队判断尾指针是否为空，为空则尾指针指向该节点，否则插入尾指针之后，转移尾指针，出队首先判断是否为空，再判断大小是否为一，为一则删除该节点，否则删除尾指针的后一个节点

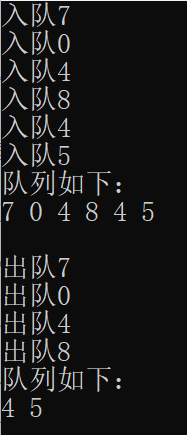
1. 算法复杂度分析

时间复杂度为O(1)，空间复杂度为O(1)

1. 实现过程中遇到的问题与解决方法

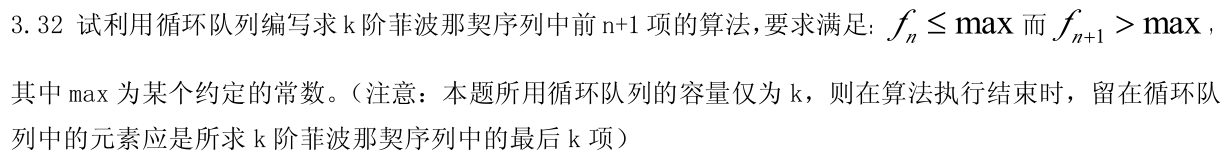
没有问题

运行结果：



结果分析：与预期相同

题目：



算法思想：

1. 算法思想与设计思路

在生成队列前k项后，将前k项加在一起得到第k+1项，然后将队列中第一项出队，将第k+1项入队

1. 算法复杂度分析

时间复杂度为O(N),空间复杂度为O(K)

1. 实现过程中遇到的问题与解决方法

没有问题

运行结果：



结果分析：与预期相同

二、代码行数及小结

代码行数为401行

了解了矩阵的列序遍历转置和快速转置算法，提高了对队列和栈的处理能力