**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：实现二叉排序树的插入和删除

算法思想：

1. 算法思想与设计思路

插入：先查找，如果找到就不插入，如果找不到，那么比较大小后，插入到parent的左或右孩子中；

删除：先查找，找到节点p后分情况：若孩子数为0或1，那么让parent指向p的孩子，若孩子数为2，在左子树中找最大元素来替代p，最后删除节点p。

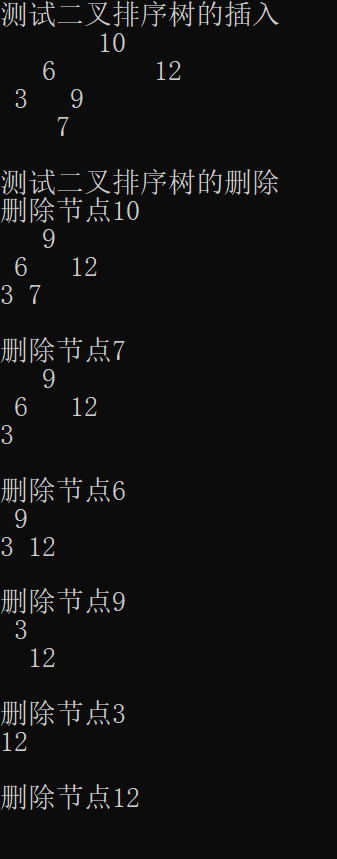
1. 算法复杂度分析

时间复杂度为O(logN),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

2、

题目：实现交换、选择、归并等简单排序算法

算法思想：

1. 算法思想与设计思路

交换排序：每次比较相邻的元素，如果不满足规定的大小关系，就交换位置

选择排序：维护一个有序部分，不断把非有序部分的最值放到有序部分尾部，当非有序部分没有元素了，就排序完毕

归并排序：采用分而治之的思想，将数组分为很多个小序列，小序列相互排序两两结合，最终得到整个有序的数组

1. 算法复杂度分析

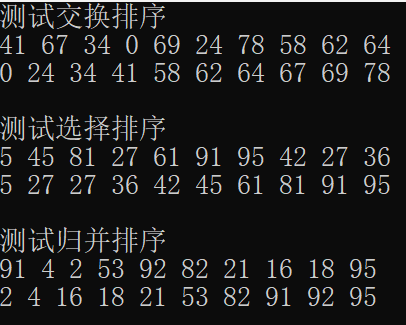
交换排序：时间复杂度为O(N2)，空间复杂度为O(1)

选择排序：时间复杂度为O(N2)，空间复杂度为O(1)

归并排序：时间复杂度为O(NlogN)，空间复杂度为O(N)

（3）实现过程中遇到的问题与解决方法

运行结果：



结果分析：与预期相同

3.

题目：实现快速排序算法

算法思想：

1. 算法思想与设计思路

从数组中选择一个数作为基准数，将大于基准数的元素都移至枢轴右边，将小于等于基准数的元素都移至枢轴左边，再对左右的子区间重复第二步的划分操作，直至每个子区间只有一个元素。

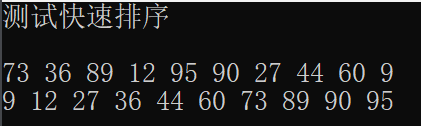
1. 算法复杂度分析

时间复杂度为O(NlogN),空间复杂度为O(logN)

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：实现堆排序算法

算法思想：

1. 算法思想与设计思路

将待排序的序列建成大根堆，此时整个序列最大值即为堆顶元素，将其与末尾元素交换，使末尾元素为最大值，然后再调整堆顶元素使得剩下的n-1个元素仍为大根堆，再重复执行以上操作得到一个有序的序列。

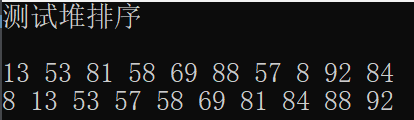
1. 算法复杂度分析

时间复杂度为O(NlogN),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

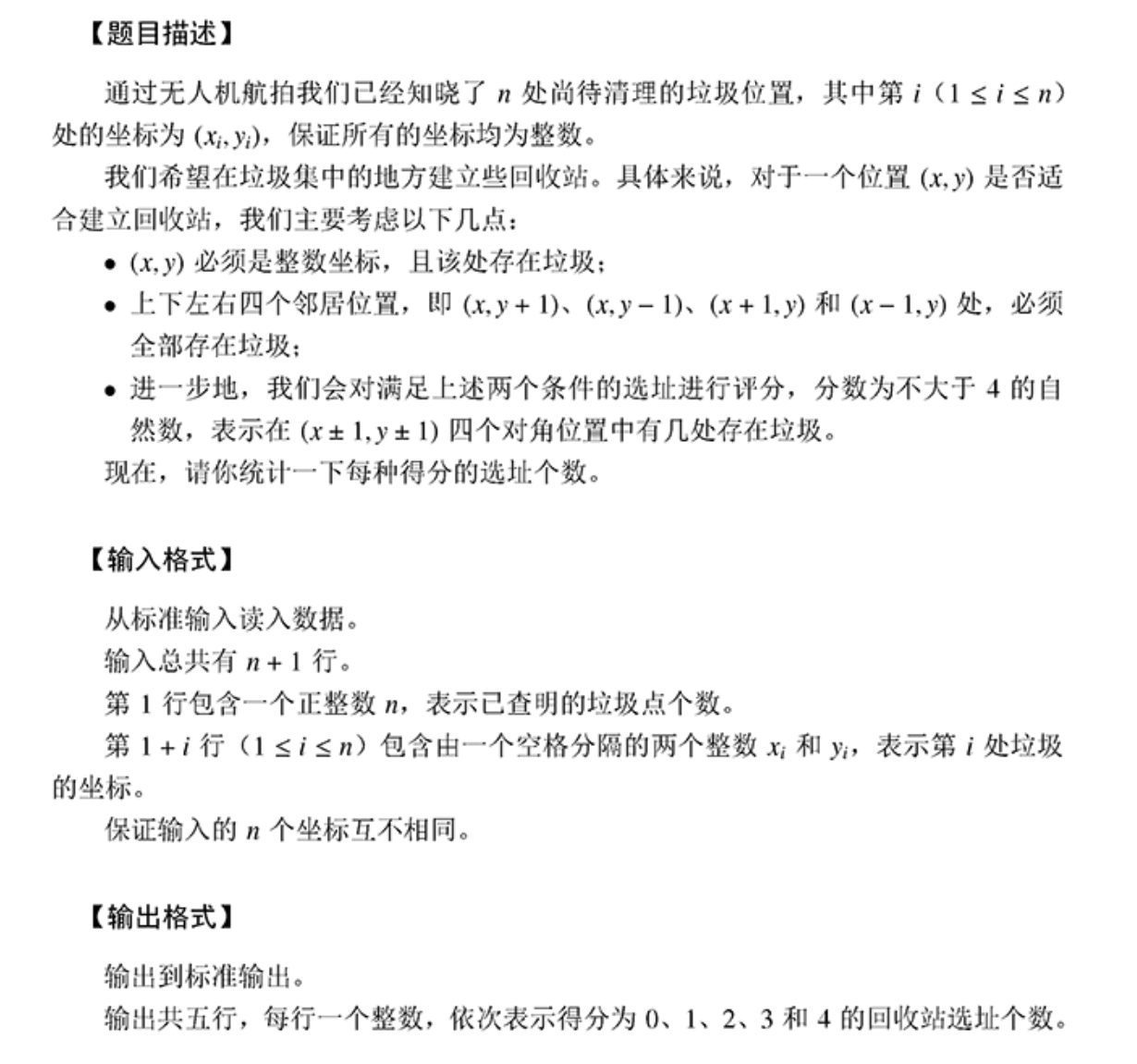
运行结果：



结果分析：与预期相同

5.

题目：



算法思想：

1. 算法思想与设计思路

采用pair来记录垃圾点的坐标,为了快速找到对应点的坐标，采用map设置key值为pair存储的坐标，value值为1,表示存在该点，直接遍历判断得到每个得分点的数量

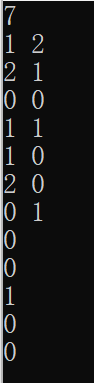
1. 算法复杂度分析

时间复杂度为O(N),空间复杂度为O(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

附加题：

T9.32

题目：已知一棵二叉排序树上所有关键字中的最小值为-max，最大值为max，又-max<x<max。编写递归算法，求该二叉排序树上的小于x且最靠近x的值a和大于x且最靠近x的值b。

算法思想：

1. 算法思想与设计思路

查找x,在查找过程中记录更新离x最近的左祖先和右祖先节点值作为a和b，若找到x，则根据其左右子树中的最大最小值是否存在来更新a和b，从而得到小于x且最靠近x的值a和大于x且最靠近x的值b

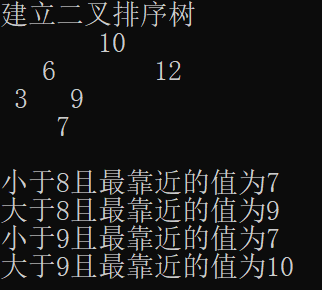
1. 算法复杂度分析

时间复杂度为O(logN),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

T10.32

题目：荷兰国旗问题：设有一个仅由红、白、蓝三种颜色的条块组成的条块序列。请编写一个时间复杂度为O(n)的算法，使得这些条块按红、白、蓝的顺序排好，即排成荷兰国旗图案。

算法思想：

1. 算法思想与设计思路

设置三个指针l,r,j,l之前元素为红色，r之后元素为蓝色，j从头开始遍历，根据j的颜色将其交换到前部或后部并移动指针，直到大于r为止

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

T10.34

题目：已知(k1,k2,…,kp)是堆，则可以写一个时间复杂度为O(logn)的算法将(k1,k2,…,kp,kp+1)调整为堆。试编写“从p=1起，逐个插入建堆”的算法，并讨论由此方法建堆的时间复杂度。

算法思想：

1. 算法思想与设计思路

从堆底部插入新元素并将其与父节点比较，若大于则交换上移，逐个插入从而构建起堆

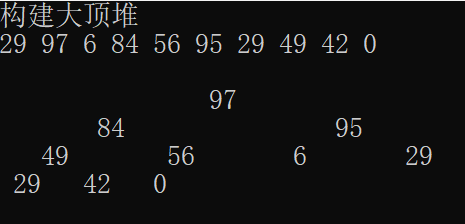
1. 算法复杂度分析

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

二、代码行数及小结

代码行数737行

了解各种各样的查找和排序的方法，对树和堆这种数据结构有了更多的理解，快速排序和归并排序的思想值得我学习