一、调试成功程序及说明

1. 实现二叉排序树的插入和删除。

算法思想：

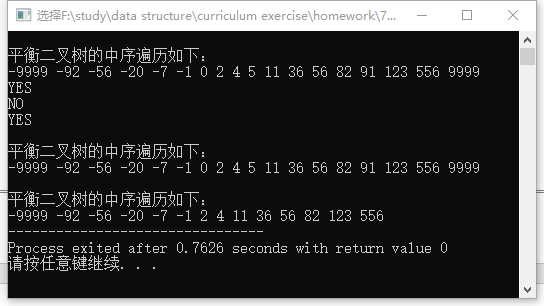
插入：

1. 寻找数据插入位置。若小于当前根结点，将其递归插入至左子树，若大于，将其递归插入至右子树，若等于则插入。
2. 向父结点回溯。若当前二叉树不平衡，回溯至最小子树并调整其至平衡。

删除：

1. 寻找应当删除的结点。若小于当前根结点，则在左子树中寻找，若大于，将在右子树中寻找，直至找到，执行步骤2。
2. 若该结点为叶结点，执行步骤4。若非叶节点，执行步骤3。
3. 若其左子树比右子树高，将该结点数据与左子树最大结点的数据交换，再删除左子树的最大结点。若右子树比左子树高同理。
4. 删除该叶结点。
5. 向父结点回溯。若当前二叉树不平衡，回溯至最小子树并调整其至平衡。

运行结果：



结果分析：

T(n) = O()

S(n) = O()

附源程序。

2. 实现交换、选择、归并等简单排序算法；

算法思想：

冒泡排序：

1. 第i趟冒泡从前往后遍历数组，若前一元素大于后一元素则交换两元素，最终使得[0,n-1)中最大元素放入n-i的位置上。
2. 重复上述操作n-1次。

选择排序：

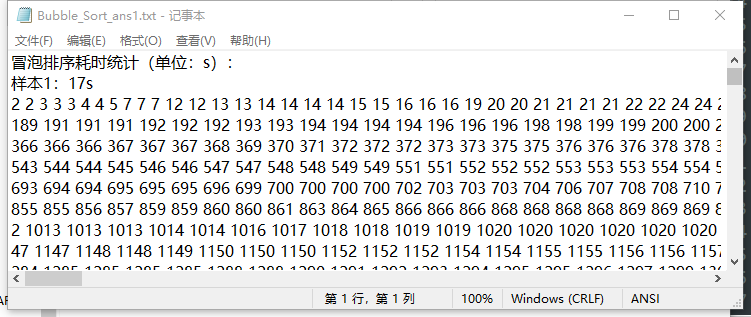
1. 选出无序序列最大值
2. 将无序序列最大值与无序序列最后位置元素互换
3. 重复上述操作n-1次。

归并排序：

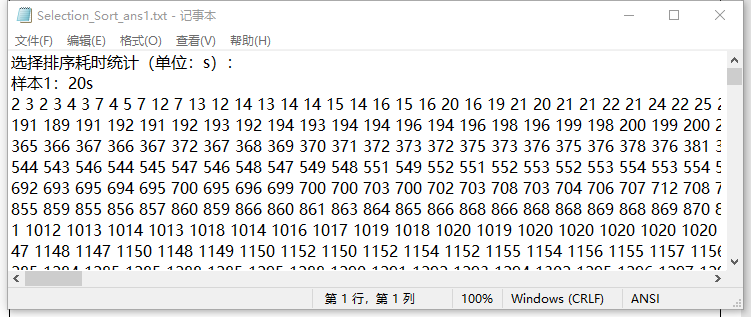
先将[left,center]和[center+1,right]归并为有序子列，再将[left,right] 归并为有序子列，递归调用以上函数。

运行结果：

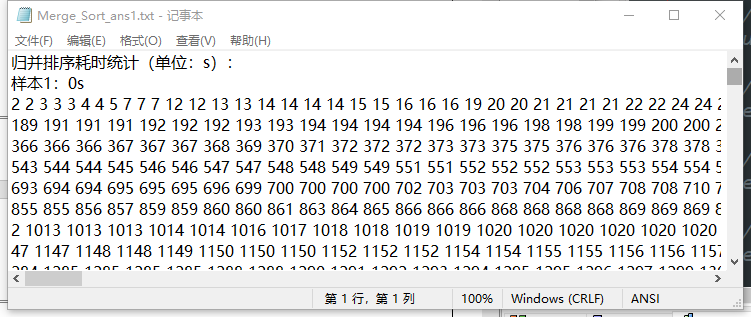
冒泡排序：



选择排序：



归并排序：



结果分析：

冒泡排序：

T(n) = O()

S(n) = O(1)

选择排序：

T(n) = O()

S(n) = O(1)

归并排序：

T(n) = O()

S(n) = O(n)

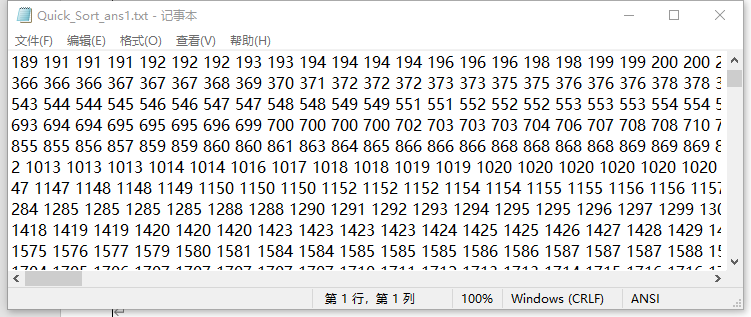
附源程序。

3. 实现快速排序算法；

算法思想：

每次将主元放在数列中间，递归调用函数，分而治之。

运行结果：



结果分析：

T(n) = O()

S(n) = O()

附源程序。

4. 实现堆排序算法；

算法思想：

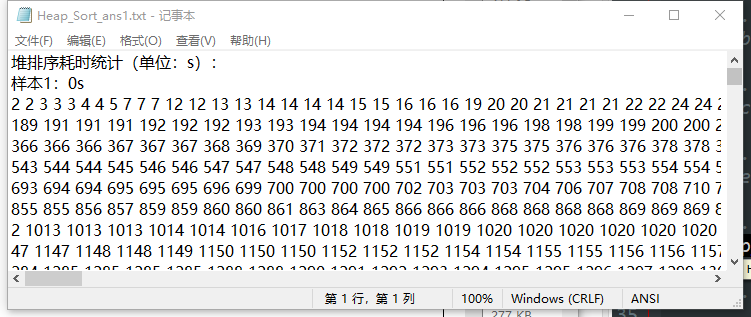
1. 建堆

可以理解为递归思想，要想把root调整为堆，要先把root的左右子树都调整为堆，再把root向下过滤，直至找到合适的位置为止。

1. 排序

每次将使大顶堆出队一个元素，将其与堆最后一个元素互换，使得堆末尾后为有序序列，再将root调整为堆，重复上述操作n-1次即可。

运行结果：



结果分析：

T(n) = O()

S(n) = O(1)

附源程序。

5． CSP题目

题目背景：

开学了，可是校园里堆积了不少垃圾杂物。

热心的同学们纷纷自发前来清理，为学校注入正能量～

题目描述：

通过无人机航拍我们已经知晓了n处尚待清理的垃圾位置，其中第i (1≤i≤n)处的坐标为(xi,yi)，保证所有的坐标均为整数。

我们希望在垃圾集中的地方建立些回收站。具体来说，对于一个位置(x,y)是否适合建立回收站，我们主要考虑以下几点：

1. (x,y)必须是整数坐标，且该处存在垃圾；
2. 上下左右四个邻居位置，即(x,y+1)、(x,y-1)、(x+1,y)和(x-1,y)处，必须全部存在垃圾；
3. 进一步地，我们会对满足上述两个条件的选址进行评分分数为不大于4的自然数，表示在(x±1,y±1)四个对角位置中有几处存在垃圾。

现在，请你统计一下每种得分的选址个数。

输入格式：

从标准输入读入数据。

输入总共有n+1行。

第1行包含一个正整数n，表示已查明的垃圾点个数。

第1+i行（1≤i≤n）包含由一个空格分隔的两个整数xi和yi，表示第i处垃圾的坐标。

保证输入的n个坐标互不相同。

输出格式：

输出到标准输出。

输出共五行，每行一个整数，依次表示得分为0、1、2、3和4的回收站选址个数。

样例1输入：

7

1 2

2 1

0 0

1 1

1 0

2 0

0 1

样例1输出：

0

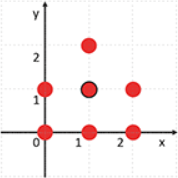
0

1

0

0

样例1解释：



如图所示，仅有(1,1)可选为回收站地址，评分为2。

样例2输入：

2

0 0

-100000 10

样例2输出：

0

0

0

0

0

样例2解释：

不存在可选地址。

样例3输入：

11

9 10

10 10

11 10

12 10

13 10

11 9

11 8

12 9

10 9

10 11

12 11

样例3输出：

0

2

1

0

0

样例3解释：

1分选址：(10,10)和(12,10);

2分选址：(11,9)。

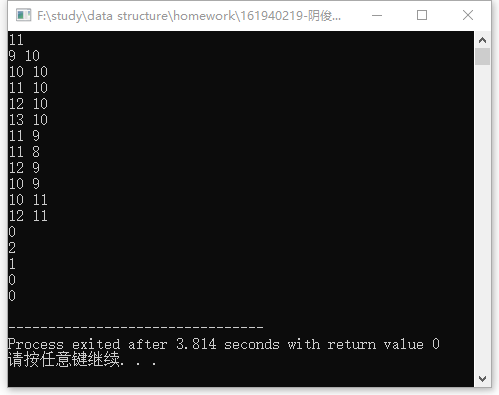
提示：

测试点1和2，保证对于任意的i皆满足0<=xi, yi<=2；  
测试点3、4和5，保证对于任意的i皆满足0<=xi, yi<= 500；  
测试点6、7和8，保证对于任意的i皆满足0<=xi, yi<= 10^9；  
测试点9和10，保证对于任意的i皆满足|xi|, |yi|<=10^9，即坐标可以是负数。  
所有的测试点保证1<=n<=10^3。

算法思想：

利用set存储垃圾图，实现结点间的高效查找。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(n)

附源程序。

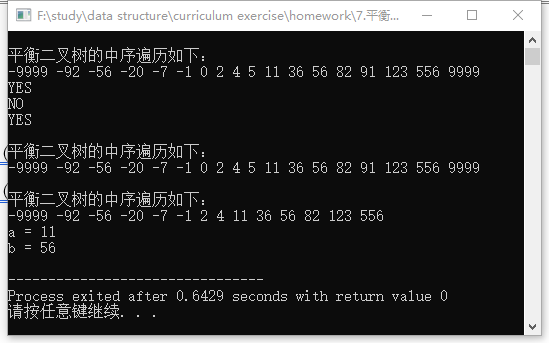
-------------------------------------------------------------------------

选做题**9.32** 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. 若当前根结点数据小于x，更新a，再去当前根结点右子树中递归寻找，若当前根结点数据大于x，更新b，去当前根结点左子树中递归寻找。
2. 若当前根结点数据等于x，若其有左子树，a更新为其左子树的最大值。若其有右子树，b更新为其右子树的最大值。

运行结果：



结果分析：

T(n) = O()

S(n) = O()

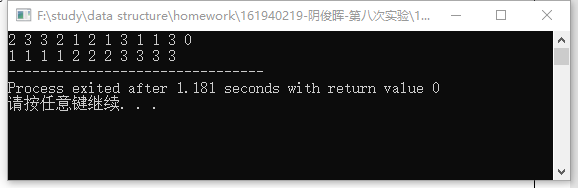
附源程序。

# 选做题10.32 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：

遍历数组，若当前指针指向元素为1，将其与1序列末尾元素互换，指针指向下一元素。若为2，指针指向下一元素。若为3，将其与3序列开头元素互换。

运行结果：



结果分析：

T(n) = O(n)

S(n) = O(1)

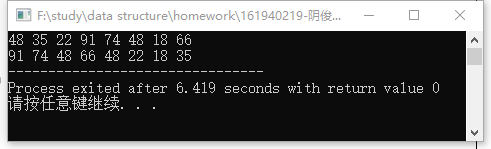
附源程序。

选做题**10.34 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. 将新插入元素放入数组末尾
2. 只要当前结点（新插入元素）大于父结点，就与父结点交换，直至当前结点小于等于父结点为止

运行结果：



结果分析：

T(n) = O()

S(n) = O(1)

附源程序。

二、代码行数及小结

|  |  |
| --- | --- |
| 题目 | 代码行数 |
| 1 + 选9.32 | 56+50+290=396 |
| 排序大法好 | 41+30+468=539 |
| 选 10.32 | 42 |
| 选10.34 | 51 |
| 总计 | 1028 |

