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：

//插入

bstnode \*Insert(bst t, int key)

{

if(!t)

{

t = (bstnode \*) malloc(sizeof(bstnode));

if(!t) exit(0);

t->data = key;

t->left = NULL;

t->right = NULL;

}

else if(t->data == key)

;

else if(t->data < key)

t->right = Insert(t->right, key);//插入右子树

else

t->left = Insert(t->left, key);//插入左子树

return t;

}

//删除

查找key所在结点p（待删结点）

//删除结点p

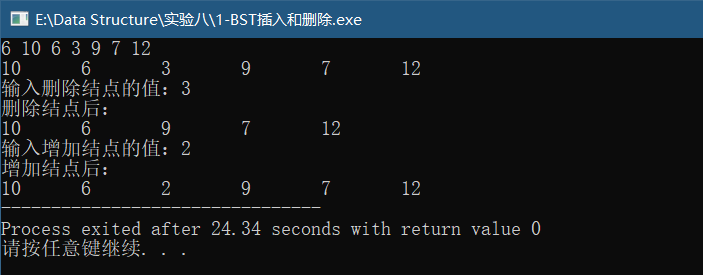
//情况一：p为叶子结点

//情况二：待删结点只有右孩子

//情况三：待删结点只有左孩子

//情况四：待删结点既有左孩子又有右孩子

运行结果：



结果分析：

附源程序。

2、

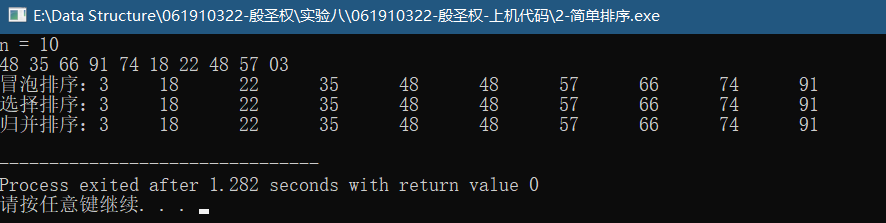
题目：

实现交换、选择、归并等简单排序算法。

算法思想：

1. 交换（冒泡）：进行n-1次冒泡，每次将最大值交换至最后。
2. 选择：每次选择表中的最小值交换至左边的有序序列，直至全部有序。
3. 归并：将前后两个相邻的有序序列归并成一个有序序列，初始状态将一个元素看成一个游戏序列，两两归并。重复归并直至归并完成。

运行结果：



结果分析：

1. 冒泡：O（n^2）
2. 选择：O（n^2）
3. 归并：O（n log n）

附源程序。

3、

题目：

实现快速排序算法。

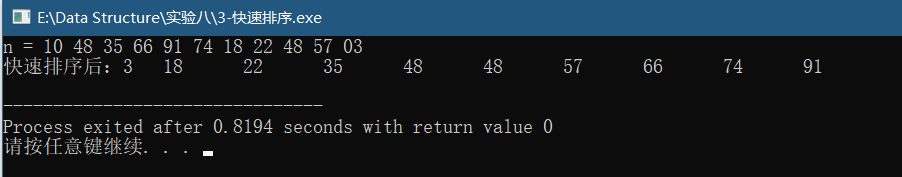
算法思想：

1、选择第一个记录为枢轴；

2、将比枢轴小的交换至枢轴左边，比枢轴大的交换至右边；

3、分别再对左右两个子序列进行1、2步直至子序列个数为1。

运行结果：



结果分析：

O（n log n）

附源程序。

4、

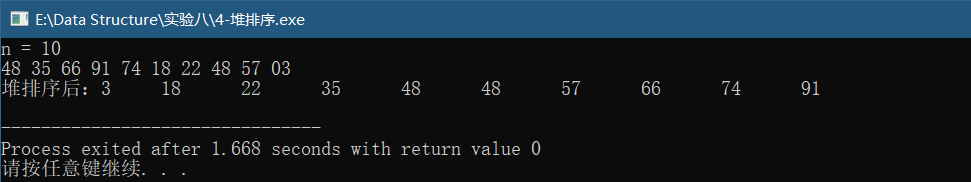
题目：

实现堆排序算法。

算法思想：

1. 由一组无序序列建成堆；
2. 将堆顶元素与表末元素交换，并在堆顶进行堆调整；
3. 重复步骤2直至全部有序。

运行结果：



结果分析：

O（n log n）

附源程序。

5、

题目：

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

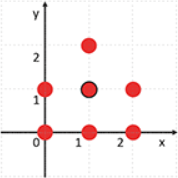
0

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)。

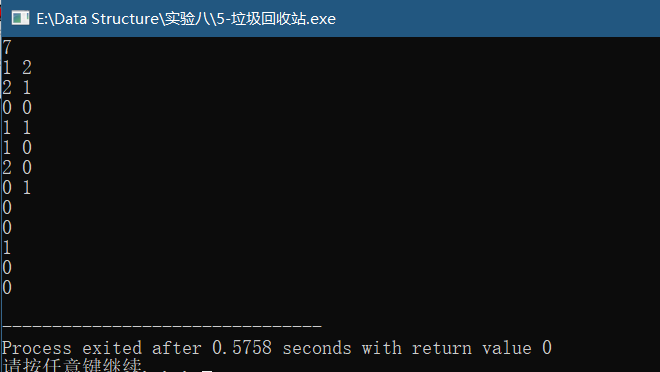
提示：

本题中所涉及的坐标皆为整数，且保证输入的坐标两两不同。

算法思想：

1. 判断是否上下左右都有垃圾
2. 若是，则判断四个角的垃圾数

运行结果：



结果分析：

O（n）

附源程序。

6、

题目：

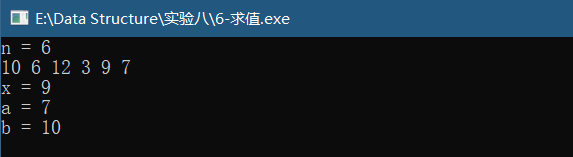
选做题9.32

已知一棵二叉排序树上的所有关键字中的最小值为-max，最大值为max，x小于最大值，大于最小值。编写递归算法，求该树上小于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：

由于二叉排序树经过中序遍历输出后为升序序列，故可基于中序遍历，a和b分别为该升序序列x左边和右边的值。

运行结果：



结果分析：

O（n）

附源程序。

7、

题目：

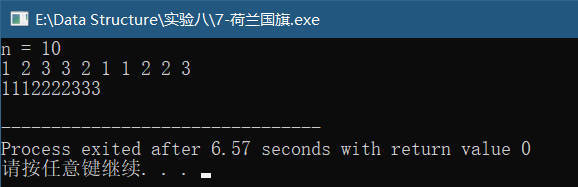
选做题10.32

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。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：

另设一个数组存储排序完成的条块序列，从两端开始先排红和蓝，红和蓝排完之后剩下的即为白色条块，再填充白色条块即可。

运行结果：



结果分析：

O（n）

附源程序。

8、

题目：

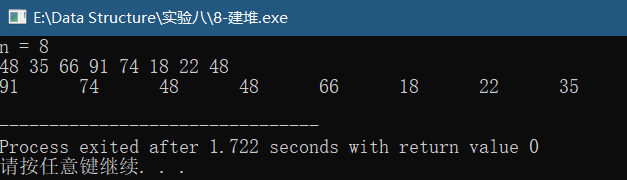
选做题10.34

已知（k1 k2 … kp）是堆，则可以写一个时间复杂度为O（log n）的算法将（k1 k2 … kp kp+1）调整成堆。试编写“从p=1起，逐个插入建堆”的算法，并讨论时间复杂度。

算法思想：

从第2个元素开始，依次往上筛选，较大者往上移。

运行结果：



结果分析：

O（n log n）

附源程序。

二、代码行数及小结

代码行数：

1、198

2、

3、66

4、65

5、62

6、68

7、37

8、56

小结：

了解并掌握了查找和排序的相关算法，能够加以运用。

能够根据各种查找和排序算法的时间复杂度和空间复杂度寻找最合适的算法并应用。