# 习题5 查找

5-1 设顺序表的长度为30，平均分成5块，每块6个元素。如果采用分块查找，则其平均查找长度为(   **C** )。

(A) 5

(B) 5.7

(C) 6.5

(D) 8.2

5-2 将关键字2，4，6，8，10，12，14，16依次存放于一维数组A[0...7]中，如果采用折半查找方法查找关键字，在等概率情况下查找成功时的平均查找长度为( **A** )。

(A) 21/8

(B) 7/2

(C) 4

(D) 9/2

5-3 简单描述静态查找和动态查找的区别。

**静态查找：指在查找过程中仅仅执行“查找”操作的查找，即查勘某特定的关键字是否在表中；检索某特定关键字数据元素的各种属性，不对表中的数据元素和表的结构进行任何改变。**

**动态查找：相较于静态查找，动态查找会多两个动作：1.首先也有一个“判断性查找”的过程，如果某特定的关键字在表中不存在，则按照一定的规则将其插入表中；2.若关键字已经存在，则可以对其进行删除操作。即动态查找的过程多了对表的操作。**

5-4 设数组A中只存放正数和负数。试设计算法，将A中的负数调整到前半区间，正数调整到后半区间。分析算法的时间复杂度。

void Arrange(int A[],int n) {

//定义双指针分别从前面与后面遍历数组

    int i=0,j=n-1,x;

    while(i<j)

    {

        while(i<j && A[i]>0)

            i++;

        while(i<j && A[j]<0)

            j--;

        if(i<j)

        {

            x=A[i];

            A[i++]=A[j];

            A[j--]=x;

            //变换A[i]与A[j]

        }

    }

}

5-5 按照“逐点插入方法”建立一个二叉排序树，树的形状取决于( **B** )。

(A) 数据序列的存储结构

(B) 数据元素的输入次序

(C) 序列中的数据元素的取值范围

(D) 使用的计算机的软、硬件条件

5-6 用利用逐点插入法建立序列(50, 72, 43, 85, 75, 20, 35, 45, 65, 30)对应的二叉排序树以后，查找元素35要在元素间进行( **B** )次比较。

(A) 3

(B) 4

(C) 5

(D) 8

5-7 给定n个整数，设计算法实现：

(1) 构造一棵二叉排序树；

(2) 从小到大输出这n个数。

//定义二叉排序树的节点结构

struct TreeNode {

    int data;

    TreeNode\* left;

    TreeNode\* right;

    TreeNode(int value) : data(value), left(nullptr), right(nullptr) {}

};

//二叉排序树节点插入函数

void insertNode(TreeNode\*& root, int value) {

    if (root == nullptr) {

        root = new TreeNode(value);

    } else {

        if (value < root->data) {

            insertNode(root->left, value);

        } else {

            insertNode(root->right, value);

        }

    }

}

//中序遍历

void inOrderTraversal(TreeNode\* root) {

    if (root != nullptr) {

        inOrderTraversal(root->left);

        cout << root->data << " ";

        inOrderTraversal(root->right);

    }

}

5-8 在平衡二叉树中，插入关键字46后得到一颗新的平衡二叉树。在新的平衡二叉树中，关键字37所在结点的左、右孩子结点中保存的关键字是( **B** )。

25

18

53

37

69

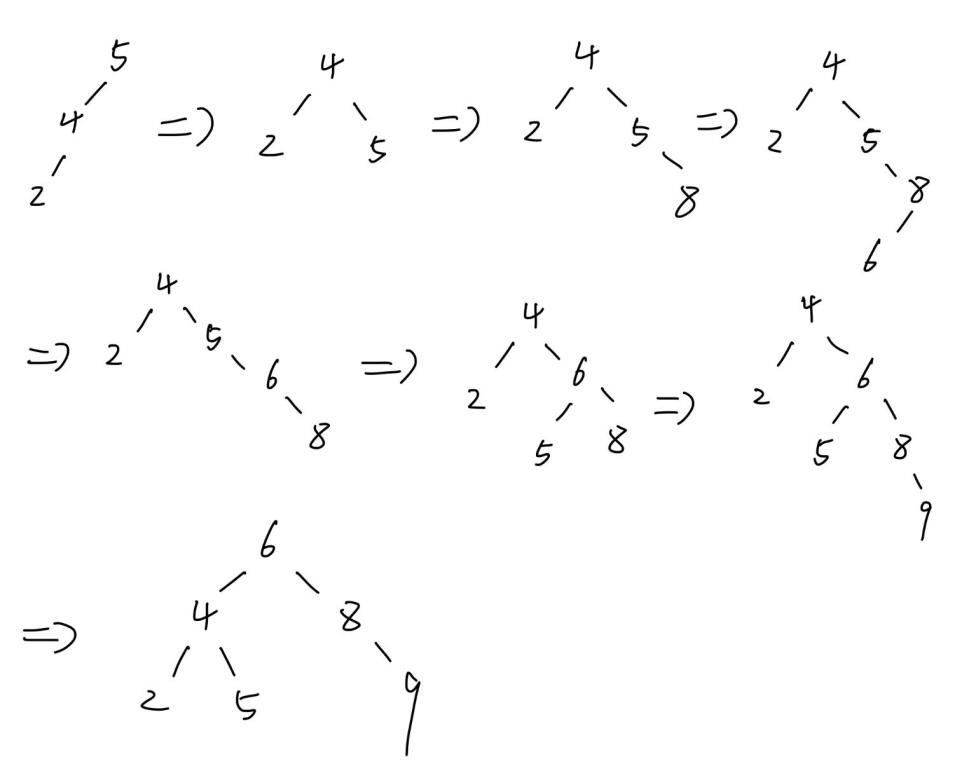
(A) 18，46

(B) 25，46

(C) 25，53

(D) 25，69

5-9 用依次插入关键字的方法，为序列{ 5, 4, 2, 8, 6, 9 }构造一棵平衡二叉树(要求分别画出构造过程中的各棵不平衡二叉树)。



5-10 链地址法是Hash表的一种处理冲突的方法，它是将所有哈希地址相同的数据元素都存放在同一个链表中。关于链地址法的叙述，不正确的是( **C** )。

(A) 平均查找长度较短

(B) 相关查找算法易于实现

(C) 链表的个数不能少于数据元素的个数

(D) 更适合于构造表前无法确定表长的情况

5-11 设哈希(Hash)函数H(k)=(3k)%11，用线性探测再散列法处理冲突，di=i。已知为关键字序列22，41，53，46，30，13，01，67构造哈希表如下：

哈希地址 0 1 2 3 4 5 6 7 8 9 10

关键字 22 41 30 01 53 46 13 67

则在等概率情况下查找成功时的平均查找长度是( **A** )。

(A) 2

(B) 24/11

(C) 3

(D) 3.5

5-12 有100个不同的关键字拟存放在哈希表L中。处理冲突的方法为线性探测再散列法，其平均查找长度为。试计算L的长度(一个素数)，要求在等概率情况下，查找成功时的平均查找长度不超过3。

素数表：101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167。

