**《数据结构与算法》作业**

# 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。

静态查找表：仅做查询和检索操作的查找表。

动态查找表：在查询之后，还需要将查询结果为不在查找表中的数据元素插入到查找表中，或者，从查找表中删除其查询结果为在查找表中的数据元素。

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) //时间复杂度为O(N)

{

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;

}

}

}

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。

typedef int KeyType;

typedef struct BiTNode

{

KeyType key; //关键字域

//ElemType\* otherinfo; //其它数据项(可以忽略)

struct BiTNode\* lchild; //左指针域

struct BiTNode\* rchild; //右指针域

struct BiTNode\* parent; //前驱节点

} BiTNode,\* BiTree;

//根据数组创建一棵二叉排序树

void create(BiTree\* root, int\* keyArray, int length)

{

int i;

for (i = 0; i < length; i++)

insert(root, keyArray[i]);

}

void insert(BiTree\* root, int data)

{

//初始化插入的节点

BiTree p = (BiTree)malloc(sizeof(BiTNode));

p->key = data;

p->lchild = NULL;

p->rchild = NULL;

p->parent = NULL;

//空树时，直接为根结点

if (!\*root)

{

\*root = p;

return;

}

//插入到当前结点的左孩子

if (!(\*root)->lchild && (\*root)->key > data)

{

p->parent = (\*root);

(\*root)->lchild= p;

return;

}

if (!(\*root)->rchild && (\*root)->key < data)

{

p->parent = (\*root);

(\*root)->rchild = p;

return;

}

if ((\*root)->key > data)

insert(&(\*root)->lchild, data);

else if ((\*root)->key < data)

insert(&(\*root)->rchild, data);

else

return;

}

void Print(BiTree root) //中序遍历打印即为从小到大

{

if (root) {

Print(root->lchild);

printf("%d ", root->key);

Print(root->rchild);

}

}

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( C )。

25

18

53

37

69

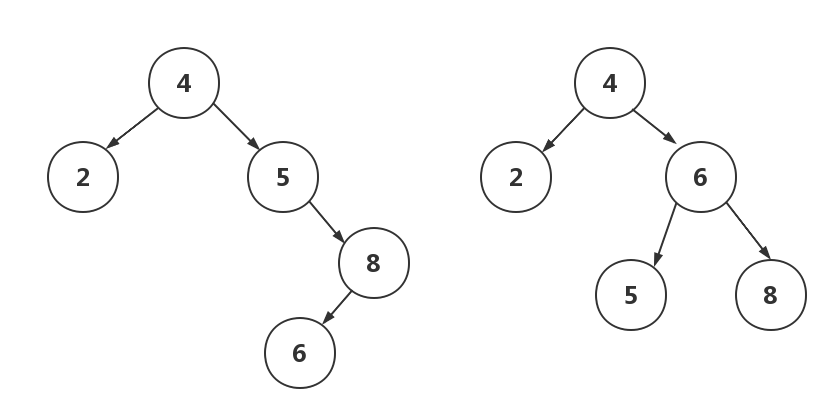
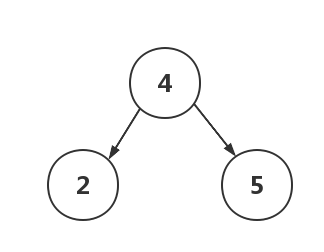
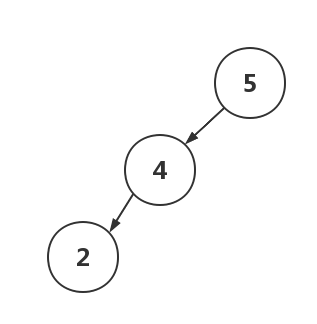
(A) 18，46

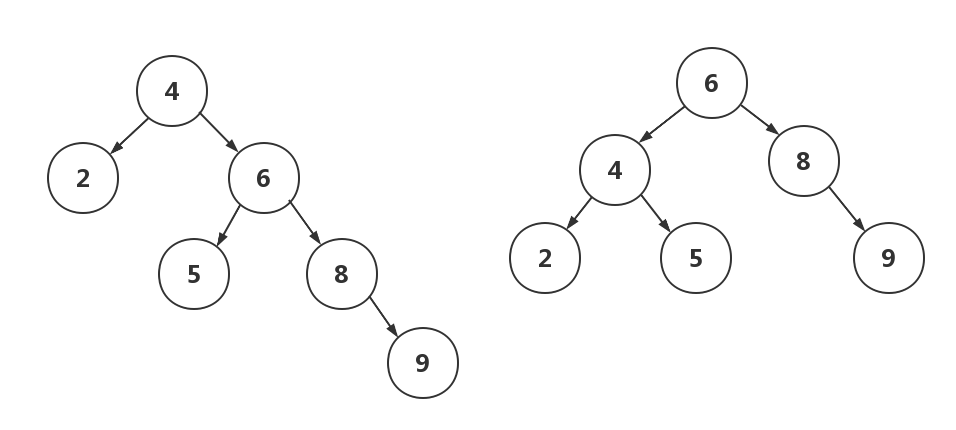
(B) 25，46

(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( B )。

(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。

<=3,解得装填因子<=0.8,L>=125,所以选取L=127.