第10次作业

## 第7章习题

**7.3构造有20个元素的二分查找的判定树，并求解下列问题：**

**（1）各元素的查找长度最大是多少？**

查找长度 查找元素

1 A[10]

2 A[5]、A[15]

3 A[2]、A[7]、A[12]、A[18]

4 A[1]、A[3]、A[6]、A[8]、A[11]、A[13]、A[16]、A[19]

5 A[4]、A[9]、A[14]、A[17]、A[20]

**（2）查找长度为1、2、3、4、5的元素各有多少？具体是哪些元素？（假设下标从0开始）**

1 A[9]

2 A[4]、A[14]

3 A[1]、A[6]、A[11]、A[17]

4 A[0]、A[2]、A[5]、A[7]、A[10]、A[12]、A[15]、A[18]

5 A[3]、A[8]、A[13]、A[16]、A[19]

**（3）查找第13个元素依次要比较哪些元素？**

查找第 13 个元素依次要比较元素：A[10]->A[15]->A[12]->A[13]

**7.5设计算法以构造有n个元素（下标范围从1到n）的二分查找判定树。**

void bstFromArr(btNode \*&T, elementType A[], int low, int high)

{

int mid;

if(low<=high)

{

mid=(low+high)/2; //计算子表中间元素下标

T=new btNode; //申请新结点并赋给数组中间元素值

T->data=A[mid];

T->lChild=NULL;

T->rChild=NULL;

bstFromArr(T->lChild, A, low, mid-1); //递归创建 T 的左子树 bstFromArr(T->rChild, A, mid+1, high); //递归创建 T 的右子树

}

}

**7.8设计算法，对给定的二叉排序树，求出在等概论情况下的平均查找长度。**

void inSum( btNode\* T, int lev, int &n, int &sum )

{

if(T)

{

inSum(T->lChild, lev+1, n, sum); //累计左子树

n++; //累计结点数

sum=sum+lev; //累计结点层次数

inSum(T->rChild, lev+1, n, sum); //累计右子树

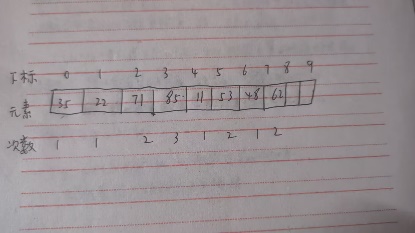
}

}

**7.12已知一棵二叉排序树如下，其各结点的值虽然未知，但其中序序列为1,2,3,4,5,6,7,8,9。请标注各结点的值**。

图7-1 题7.12图

**7.13已知散列表地址区间为0～9,散列函数为H(k)=k % 7,采用线性探测法处理冲突。将关键字序列11,22,35,48,53,62,71,85依次存储到散列表中，试构造出该散列表，并求出在等概论情况下的平均查找长度。**



平均查找长度 ASL=(1+1+2+3+1+2+1+2)/8=13/8。

**7.17已知散列表的地址区间为0～10，散列函数为H(k)=k % 11，采用线性探测法处理冲突。设计算法在其中查找值为x的元素，若查找成功，返回其下标，否则返回－1。**

int Search(elementType A[], keyType x)

{

int p,h,i;

int sn; //保存查找次数

if(n==0) //n 为表中实际元素数

return -1; //空表，查找失败

h=x%p; //计算待查关键字 x 的哈希值，p=11

if(A[h].key==x)

return h; //查找成功，返回元素下标

else

{

i=(h+1)%MaxLen; //从下一个元素开始线性探测查找，MaxLen=11

while(i!=h)

{

if(A[i].key==x) //探测查找成功，返回元素下标

return i;

i=(i+1)%MaxLen; //否则，循环线性探测，MaxLen=11

}

return -1; //元素不在表中

}

}