**第七章作业**

**（一）7.2 对有序数据表（5,7,9,12,15,18,20,22,25,30,100），按二分查找方法模拟查找元素 10 和 28，并分别画出其搜索过程。**

**查找10，low=0,high=10,**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
| **low=0** |  |  |  |  | **mid=5** |  |  |  |  | **high=10** |

**low<high进入第一轮， mid=5, 10<18，返回high=mid-1=4,**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
| **low=0** |  | **mid=2** |  | **high=4** |  |  |  |  |  |  |

**low<high进入第二轮, mid=(low+high)/2=2 ,10>9，返回low=mid+1=3**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
|  |  |  | **low=mid=3** | **high=4** |  |  |  |  |  |  |

**low<high进入第三轮, mid=(low+high)/2=3,10<12，返回high=mid-1=2,**

**low>high返回查找失败**

**查找28, low=0,high=10,**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
| **low=0** |  |  |  |  | **mid=5** |  |  |  |  | **high=10** |

**low<high进入第一轮， mid=5， 28>18,返回low=mid+1=6**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
|  |  |  |  |  |  | **low=6** |  | **mid=8** |  | **high=10** |

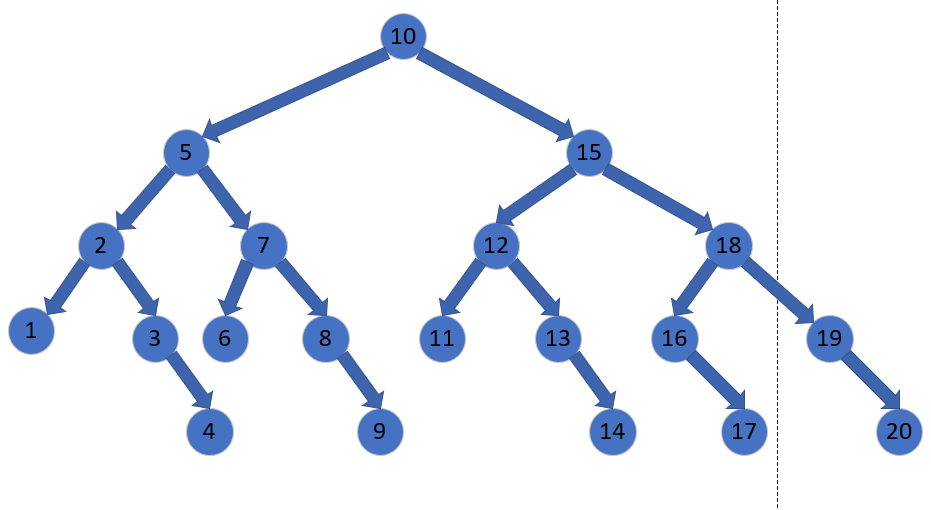
**low<high进入第二轮，mid=(low+high)/2=8， 28>25,返回low=mid+1=9；**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **5** | **7** | **9** | **12** | **15** | **18** | **20** | **22** | **25** | **30** | **100** |
|  |  |  |  |  |  | **low=6** |  | **mid=8** |  | **high=10** |

**low<high进入第三轮，mid=(low+high)/2=9,28<30，返回high=mid-1=8**

**low>high返回查找失败**

**（二）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？**

最大查找长度是5

**（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）**

查找长度是1的为下标为9的元素

查找长度是2的为下标为4，14的元素

查找长度是3的为下标为1,6,11,17的元素

查找长度是4的为下标为0,2,5,7,10,12,15,18的元素

查找长度是5的为下标为3,8,13,16,19的元素

**（3）查找第 13 个元素依次要比较哪些元素？**

依次比较10,15,12,13

**（三）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。**

11%7=4,22%7=1,35%7=0,48%7=6,53%7=4，62%7=6,71%7=1，85%7=1

散列表为

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 35 | 22 | 71 | 85 | 11 | 53 | 48 | 62 |  |  |

查找次数 1 1 2 3 1 2 1 2

查找长度=（1\*4+2\*3+3\*1）/8=13/8

**（四）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。**

算法分析：

首先根据散列表除留余数法插入方式构造链表，k作为表长，将元素取余，余数对应散列表A[k]的下标k,如果某个地址已经插入了元素，则向后搜索空位插入，

搜索时，当关键字相同时，首先获取该元素的余数，再获取其查找次数，余数+查找次数-1再除以10取余即为其下标。

**源代码：**

#include<iostream>

using namespace *std*;

typedef struct node

{

int key;

struct node\* next;

}hatable;

void SetHash(int m,int n,hatable\*a[1000])

{

for (int i = 0; i < m; i++)

{

hatable \*p = new hatable;

p->next = *NULL*;

a[i] = p;

}

*cout* << "依次输入元素" << *endl*;

for (int i = 0; i < n; i++)

{

int num;

*cin* >> num;

int address = num % m;

hatable\* tmp;

tmp = a[address];

while (tmp->next != *NULL*)//不为空，已经有元素存在，指针后移

{

tmp=tmp->next;

}

if (tmp->next == *NULL*)//为空直接插入

{

hatable\* q =new hatable;

tmp->next = q;

q->key = num;

q->next = *NULL*;

// cout << tmp->next->key<<endl;

}

}

}

int SearchHash(int m, int num,hatable\*a[])

{

int address = num % m;

hatable\* p=new hatable;

p = a[address]->*next*;

int count = 0;

while (p != *NULL*)

{

count++;//查找次数

if (num == p->key)//如果查找关键字相同

{

return (address+count-1)%10;//返回元素在表中的下标

}

p = p->next;//继续向后查找

}

return -1;

}

int main()

{

int m, n;

*cout* << "输入表长"<<*endl*;

*cin* >> m;

*cout* << "输入关键字个数" << *endl*;

*cin* >> n;

hatable\* a[1000];

SetHash(m, n,a);

int num;

*cout* << "输入要查找的元素" << *endl*;

*cin* >> num;

if (SearchHash(m, num, a)==-1)

{

*cout* << "-1";

}

else

{

*cout*<<"元素下标为"<< SearchHash(m, num, a);

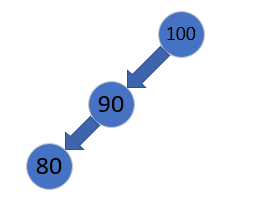
}

return 0;

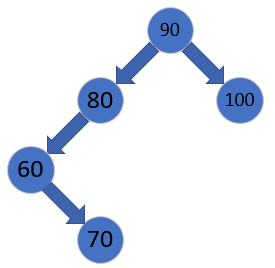
}

**（五）依次输入下列数据构造一棵平衡二叉树：（要求有过程）**

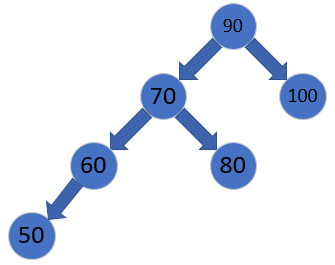
**100，90，80，60，70，50，120，110，150，87**



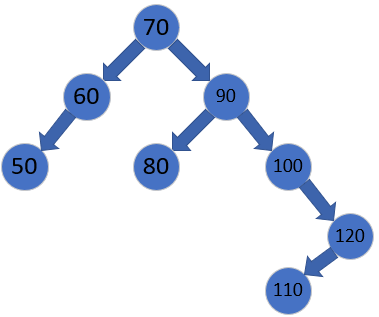
插入80不平衡做LL型调整，继续插入60,70后如下图



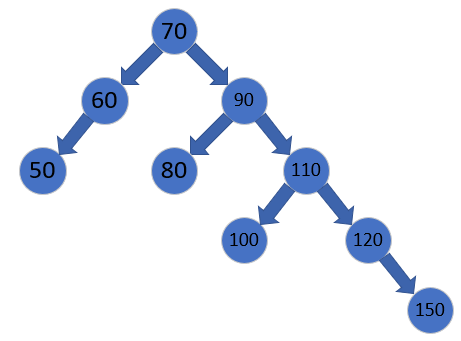
插入到70时不平衡做LR型调整，继续插入50后如下图



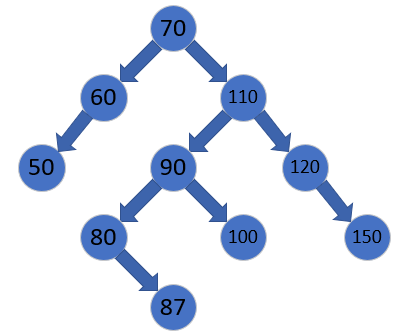
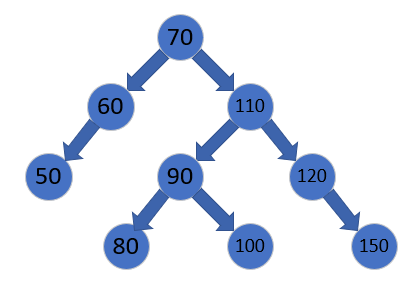
插入50后不平衡，LL型调整，继续插入120,110如下图



插入到110时不平衡，做RL型调整后，继续插入150，如下图



此时不平衡，做RR型调整，再插入87如下图



插入到87时，失去平衡，进行RL调整，调整后如下图

