

# Chapter 06

谢胡睿 2400014151 数算（B） 24-25春

作业(p199~200):

- 复习题:2、3、4、5
- 算法题:3、5

作业(p248~249):

- 复习题:1、2、3、7
- 算法题:1

## p199-200

### 复习题2.

*Ans* : 总共需要3次, 过程如下:  
首先, 字典总共16个元素, 且已经有序。  
使用二分法则先从中间开始,

1.  $l = 0, r = 15, mid = 7$   
 $D[7] = 509 < 612$ , 则  $l = 7 + 1 = 8$
2.  $l = 8, r = 15, mid = 11$   
 $D[11] = 677 > 612$ , 则  $r = 11 - 1 = 10$
3.  $l = 8, r = 10, mid = 9$   
 $D[9] = 612 == 612, finish!$

### 复习题3.

因为  $\alpha = 0.5$ , 则  $p = 16/0.5 = 32$ , 做取余哈希

$$H(key) = key \pmod{32}$$

元素	$key \pmod{32}$	地址	是否碰撞?
16	$16 \pmod{32} = 16$	16	No
87	$87 \pmod{32} = 23$	23	No
154	$154 \pmod{32} = 26$	26	No
170	$170 \pmod{32} = 10$	10	No
275	$275 \pmod{32} = 19$	19	No
426	$426 \pmod{32} = 10$	10	Yes (170)
503	$503 \pmod{32} = 23$	23	Yes (087)
509	$509 \pmod{32} = 29$	29	No
512	$512 \pmod{32} = 0$	0	No
612	$612 \pmod{32} = 4$	4	No
653	$653 \pmod{32} = 13$	13	No
677	$677 \pmod{32} = 5$	5	No

元素	$key \pmod{32}$	地址	是否碰撞?
703	$703 \pmod{32} = 31$	31	No
765	$765 \pmod{32} = 29$	29	Yes (509)
897	$897 \pmod{32} = 1$	1	No
908	$908 \pmod{32} = 12$	12	No

碰撞3次。

复习题4.

用开地址线性探查法：  
426 :  $T[11] = 426, T[10] = 170$   
503 :  $T[24] = 503, T[23] = 87$   
765 :  $T[30] = 765, T[29] = 509$   
存储状况如下：  
 $T[0] = 512$   
 $T[1] = 897$   
 $T[4] = 612$   
 $T[5] = 677$   
 $T[10] = 170$   
 $T[11] = 426$   
 $T[12] = 908$   
 $T[13] = 653$   
 $T[16] = 016$   
 $T[19] = 275$   
 $T[23] = 087$   
 $T[24] = 503$   
 $T[26] = 154$   
 $T[29] = 509$   
 $T[30] = 765$   
 $T[31] = 703$   
其余均为空

复习题5.

顺序，二分易于理解，实现简单（二分需要排序 $O(n\log n)$ ），且是离线的，不能在线处理）  
哈希法需要找到一个比较好的函数，坏的函数时间复杂度将趋近于 $O(n)$   
哈希法要消耗额外的空间  
故三种方法各有利弊，需按需完成。

算法题3.

顺序存储结构：

```

typedef struct{
    int key;
    int info;
}Node;
#define MAX 1000
typedef struct{
    Node e[MAX];
    int n;
}Seq;
int func(int key,Seq *seq){
    for(int i=0;i<seq->n;i++){
        if(seq->e[i].key==key){
            if(i){
                Node temp;
                temp=seq->e[i-1];
                seq->e[i-1]=seq->e[i];
                seq->e[i]=temp;
                return seq->e[i-1].info;
            }
            else {
                return seq->e[i].info;
            }
        }
    }
    return -1;
}

```

链式存储：

```

typedef struct {
    int data;
    struct LNode *next;
}LNode,*LList;
bool func(LList *lst,int key){
    if(*lst==NULL) return false;
    LNode *prepre=NULL;
    LNode *pre=NULL;
    LNode *crr=*lst;
    while(crr){
        if(crr->data==key){
            if(pre){
                pre->next=crr->next;
                if(prepre){
                    prepre->next=crr;
                }
            }
            else{
                *lst=crr;
            }
            crr->next=pre;
            return true;
        }
        return true;
    }
}
return false;
}

```

## 算法题5.

```
enum SlotStatus{EMPTY,OCCUPIED,DELETED};
struct HT{
    int key;
    SlotStatus status;
};
HT table[MAX];
//init every entry as EMPTY

bool Delete(int key_to_delete){
    int i=0;
    int initial_hash_val;
    int step_hash_val;
    int current_address;
    initial_hash_val=h1(key_to_delete)%m;
    step_hash_val=h2(key_to_delete)%m;//互素
    if(step_hash_val==0){
        step_hash_val=1;
    }
    do{
        current_address=(initial_hash_val+i*step_hash_val)%m;
        if(table[current_address].status==OCCUPIED&&table[current_address].key==key_to_delete){
            table[current_address].status=DELETED;
            return true;
        }
        if (table[current_address].status==EMPTY){
            return false;
        }
        i++;
    }while (i<m);//周期
    return false;
}
```

复习题1.

1. 10  
  \  
    20  
      \  
        30  
{10,20,30}

2. 10  
  \  
    30  
   /  
    20  
{10,30,20}

3. 20  
  /  
  \  
10  30  
{20,10,30} or {20,30,10}

4. 30  
  /  
  10  
   \  
    20  
{30,10,20}

5.   30  
   /  
   20  
  /  
  10  
{30,20,10}

复习题2.

即求卡特兰数 $C_4$   
 $C_4 = \sum_{i=0}^3 C_{4-i-1} * C_i$   
其中:  $C_0 = 1, C_1 = 1$   
则 $C_4 = 14$

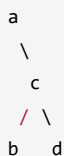
1. badc,bdca

b  
  /  
a  \  
   d  
   /  
  c

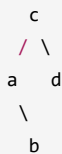
2. bacd,bcda,bcad

b  
  /  
a  \  
   c  
   \  
    d

3. acbd,acdb



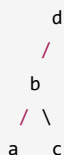
4. cabd,cdab



5. cbad,cdba



6. dbca,dbac



### 复习题3.

证明:

对称序列即为中序遍历:(左-中-右)

假设不是关键码顺序,即存在 $key(A) < key(B)$ ,使得 $A$ 在 $B$ 后面

因为 $key(A) < key(B)$ ,所以 $A$ 在 $B$ 的左子树,或者在 $B$ 的父节点/父节点的左子树中(由 $BST$ 性质得到)

1. 在 $B$ 左子树中,矛盾!(先遍历 $B$ 的左子树)
2.  $A$ 为 $B$ 的父节点,矛盾!(左-中-右)
3.  $A$ 在 $B$ 的父节点的左子树中,矛盾!

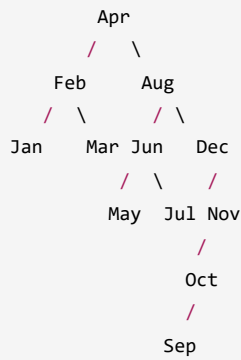
所以必然是关键码顺序

### 复习题7.

字典的字母序:

*Apr, Aug, Dec, Feb, Jan, Jun, Jul, Mar, May, Nov, Oct, Sep*

树为:



存储的关键字	对应的月份大小	深度 $(d_i)$	比较次数 $(d_i + 1)$
Apr	4	0	1
Feb	2	1	2
Aug	8	1	2
Jan	1	2	3
Mar	3	2	3
Jun	6	2	3
Dec	12	2	3
May	5	3	4
Jul	7	3	4
Nov	11	3	4
Oct	10	4	5
Sep	9	5	6
总和			40

则平均次数为10/3

### 算法题1.

```
typedef struct{
    Node* left=NULLptr;
    Node* right=NULLptr;
    int key;
}Node,*Tree;
int depth(Node *root,int key){
    int depth=0;
    Node *temp=root;
    while(temp){
        if(temp->key==key)return depth;
        if(temp->key>key){
            temp=temp->left;
            depth++;
            continue;
        }
        temp=temp->right;
        depth++;
    }
    return -1;
}
```