

数据结构导论

线性表

顺序表的结构定义

```
const int Maxsize = 100;    //预先定义一个足够大的常数
typedef struct
{
    DataType data[Maxsize];    //存放数据的数组
    int length;                //顺序表的实际长度
} SeqList;                    //顺序表类型名为SeqList
SeqList L;                    //定义L为一个顺序表
```

单链表的类型定义

```
typedef struct node //struct node 表示链表的结点，结点包含两个域：数据域 (data) 和指针域 (next)
{
    DataType data;    //数据域
    struct node * next; //指针域
} Node, *LinkList;
```

循环链表：在单链表中，如果让最后一个结点的指针域指向第一个结点可以构成循环链表。在循环链表中，从任一结点出发能够扫描整个链表。

双向循环链表

```
struct dbnode
{
    DataType data;
    struct dbnode *prior, *next;
}
typedef struct dbnode *dbpointer;
typedef struct DLinkList;
```

栈、队列、数组

顺序栈

```
const int maxsize = 6;    //顺序栈的容量
typedef struct seqstack
{
    DataType data[maxsize];    //存储栈中数据元素的数组
    int top;                    //标志栈顶位置的变量
} SeqStk;
```

链栈

```
typedef struct node
{
    DataType data;
    struct node *next;
} LkStk;
```

顺序队列

```
const int masize = 20;
typedef struct seqqueue
{
    DataType data[maxsize];
    int front, rear;
}SeqQueue;
SeqQueue SQ;
```

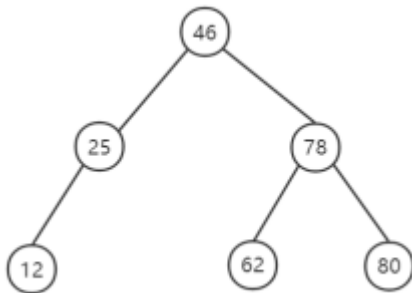
链接队列：队列的链接实现实际上是使用一个带有头结点的单链表来表示队列，称为链队列。头指针指向链表的头结点，单链表的头结点的 next 域指向队列首结点，尾指针指向队列尾结点，即单链表的最后一个结点。

```
typedef struct LinkQueueNode
{
    DataType data;
    struct LinkQueueNode *next;
} LkQueueNode;

typedef struct LKQueue
{
    LKQueueNode *front, *rear;
} LKQueue;
LKQueue LQ;
```

第 6 章 查找

1. 给定数据序列{46,25,78,62,12,80}，试按元素在序列中的次序将它们依次插入一棵初始为空的二叉排序树，画出插入完成后的二叉排序树。



2. (6 分) 给定有序表 D={006, 087, 155, 188, 220, 465, 505, 508, 511, 586, 656, 670, 700, 766}，用 二分查找法 在 D 中查找 511，试给出查找过程。

01	02	03	04	05	06	07	08	09	10	11	12	13	14
(1)006	087	155	188	220	465	505	508	511	586	656	670	700	766
↑						↑							↑
low						mid							high

(2)006	087	155	188	220	465	505	508	511	586	656	670	700	766
							↑			↑			↑
							low			mid			high

(3)006	087	155	188	220	465	505	508	511	586	656	670	700	766
							↑	↑	↑				
							low	mid	high				

3. 如图所示长度为 13 的散列表，其散列函数为 $H(\text{key}) = \text{key} \bmod 13$ ，在表中已填入键值分别为 16，30，54 的元素。

0	1	2	3	4	5	6	7	8	9	10	11	12
		54	16	30								

1) (3 分) 现要插入键值为 29 的元素，应用线性探测法，计算填入散列表中单元的序号。(要求给出求解过程)

散列函数求出其散列地址为 3，在地址 3 上面已有元素 16，发生冲突。应用线性探测法，得到下一个地址为 $d+1=4$ ，仍冲突，则再求下一个地址 $d+2=5$ ，这个位置上没有元素，将元素填入散列表中序号为 5 的单元。

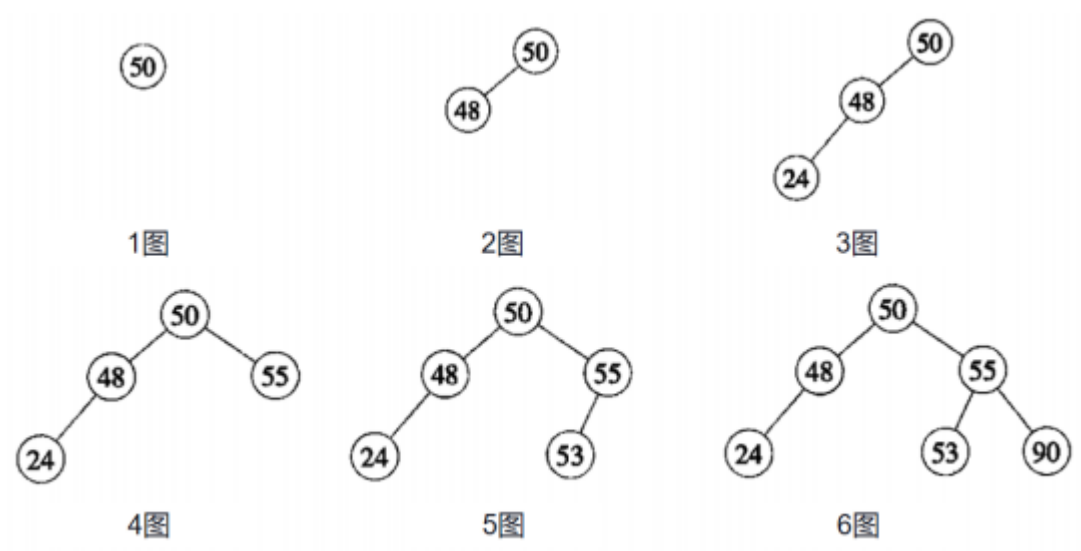
2) (3 分) 线性探测法中，如何减少堆积的机会？

应设法使后继散列地址尽量均匀地分散在整个散列表中。

4. 给定表(Jan, Feb, Mar, Apr, May, Jul)。散列表的地址空间为 0~10，设散列函数 $H(x) = \lfloor i/2 \rfloor$ ，其中 i 为键值中第一个字母在英语字母表中的序号，要求画出以线性探测法解决冲突的散列表。

0	1	2	3	4	5	6	7	8	9	10
Apr			Feb		Jan	Mar	May	Jul		

5. 根据二叉排序树的插入算法，从空树开始建立键值序列{50，48，24，55，53，90}的二叉排序树，要求给出建立过程。



排序
