数据结构与算法一

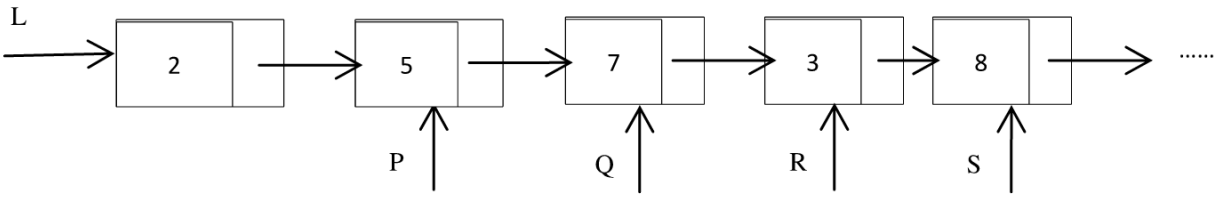
1.单链表中，设置头结点的作用是什么？

链表由头结点中的头指针唯一确定，起到标识的作用。

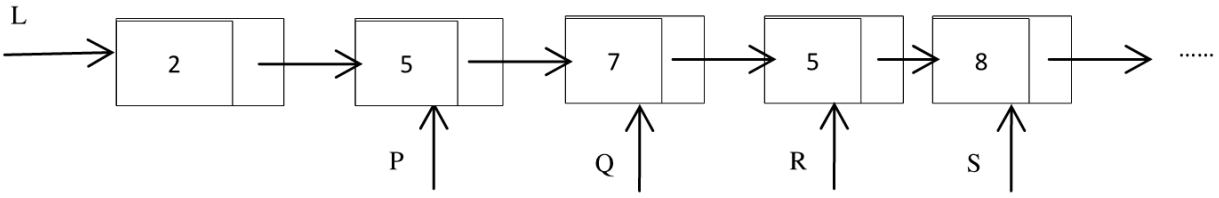
2.顺序表和链表分别在什么情况下使用合适？

小型、结构简单的数据存储用顺序表更合适，需要对存储数据进行大量操作时用链表更合适。

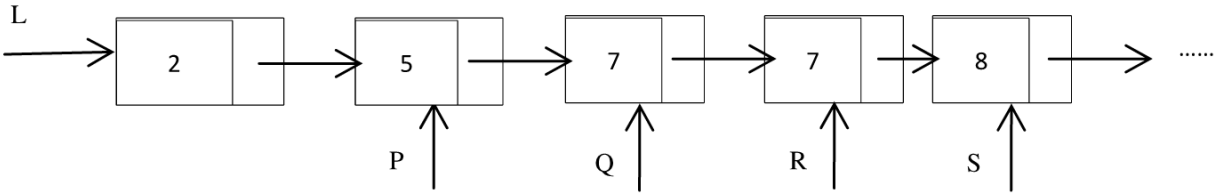
3.对下面单链表，分别执行下属各语句，各结果如何？请画出结果示意图。



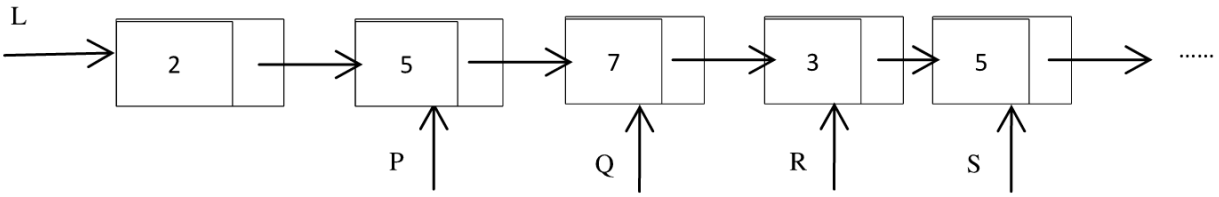
① R->data = P->data



② R->data = P->next->data

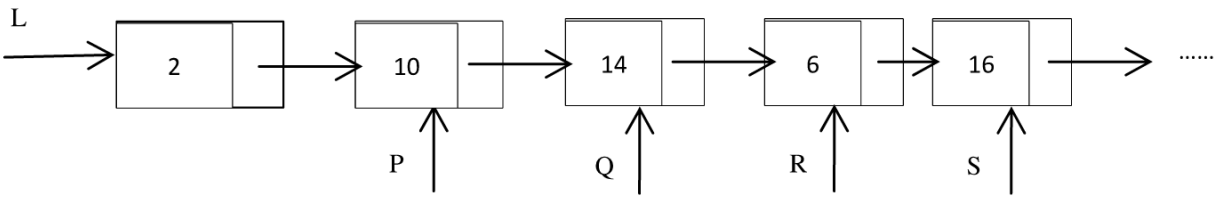


③ P->next->next->next->data=P->data



④ T=P;

while(T!=NULL) {T->data=T->data*2; T=T->next}



4. 设 P 结点是某双向链表的中间结点（非头结点和尾结点），请使用 C/C++语言写出实现下列功能的伪代码：

① 在 P 结点后插入 S 结点

```
s->former = p;
s->next = p->next;
p->next = s;
p->next->former = s;
```

② 在 P 结点前插入 S 结点

```
s->former = p->former;
s->next = p;
p->former = s;
p->former->next = s;
```

③ 删除 P 结点的直接后继结点

```
p->next = p->next->next;
p->next->next->former = p;
```

④ 删除 P 结点的直接前驱结点

```
p->former = p->former->former;
p->former->former->next = p;
```

⑤ 删除 P 结点

```
p->former->next = p->next;
p->next->former = p->former;
```

5. 已知一线性表含有 n 个数据元素，其中数据元素以值递增有序排列，并以单链表形式存储。设计一个算法来删除表中大于 Kmin 且小于 Kmax 的元素，并计算你所设计算法的时间复杂度，以 C/C++语言来表达。注意要考虑表中不存在大于 Kmin 且小于 Kmax 的元素的情况。

```
if(head->data < Kmax)
{
    for(int i = head; i != NULL; i = i->next)
    {
        if(i->data <= Kmin)
            int k1 = i;
        if(i->data < Kmax)
            int k2 = i;
    }
    k1->next = k2->next;
}
```

时间复杂度T(n)=o(n)。

数据结构与算法二

（1）栈、队、串结构习题

1. 简述栈和顺序表的区别；简述队列和栈的区别。

栈限定只能在一端进行插入和删除操作，队列限定只能在一端进行插入，另一端进行删除操作，而顺序表可以对每个元素进行插入和删除操作；队列先进先出，栈先进后出。

2. 写出下列程序段的输出结果（设栈的元素类型为SElemType为char）

```
void main ()
{
    stack S;
    char x,y;
    InitStack(S);
    x='c'; y='k';
    push(S,x); push(S,x); push(S,x); push(S,x); push(S,x); push(S,x);
    while (! StackEmpty(S))
    {
        pop(S,y); printf(y);
    }
    printf(y);
}
```

输出结果为：ccccccc

3. 简述空串与空格串的区别。

空串内只有一个空字符'\0'，空格串内元素为空格'32'。

4. 已知：s= ‘(XYZ)+*’,t= ‘(X+Z)*Y’ 。利用联接、求字串和置换等基本运算，将 s 转化为 t

```
SubString(s1, s, 0, 2);
SubString(s2, s, 2, 1);
SubString(s3, s, 3, 2);
SubString(s4, s, 5, 1);
SubString(s5, s, 6, 1);
Concat(s, s1, s4);
Concat(s, s4, s3);
Concat(s, s3, s5);
Concat(s, s5, s2);
```

5. 编写算法 ,从串 s 中删除所有和串 t 相同的字串： s='Joint Strike Fighter' ,t='fight' ，使用 C/C++ ，写出代码，画出实现流程图。

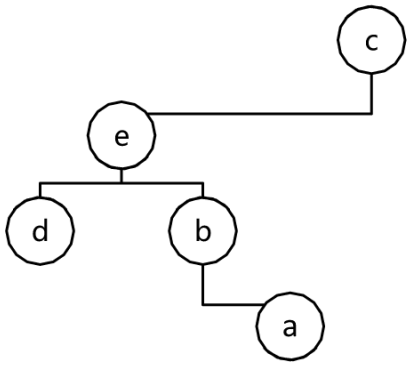
```
for(int i = 0; i <= StrLength(s)- 5; ++i)
{
    if(StrCompare(t, SubString(t1, s, i, 5)) == 0)
        Concat(s, SubString(s1, s, 0, i), SubString(s2, s, i, StrLength(s)-5));
}
```

循环体内的流程图：

（2）树结构习题

1. 已知某二叉树的后序遍历序列是 dabec。中序遍历序列是 debac，写出它的前序遍历序列。

此二叉树为

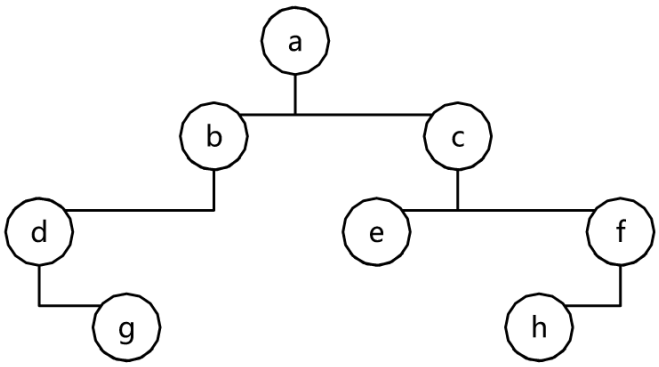


前序遍历序列为 cedba。

2.某二叉树的前序遍历结点访问顺序是 abdgcefh ,中序遍历的结点访问顺序是 dgbaechf ，写出其后序遍历结点访问顺序。

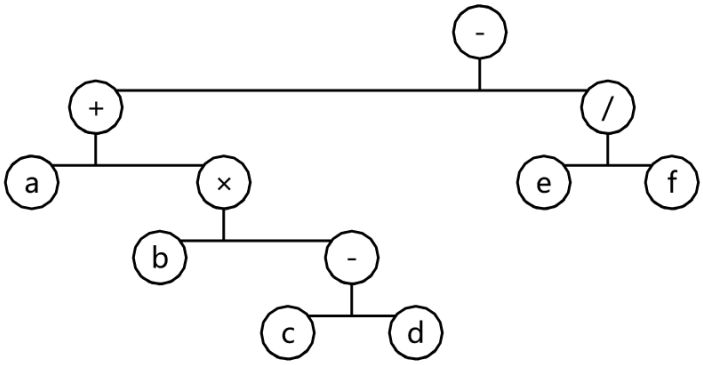
此二叉树为

后序遍历序列为 gdbehfca。



3. 请写出右图所示二叉树的先序、中序和后序遍历结果。

先序遍历结果：-+a×b-cd/ef



中序遍历结果：a+b×c-d-e/f

后序遍历结果：abcd-×+ef/-

4. 任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序（A）

（A）不发生改变　　（B）发生改变　　（C）不能确定　　（D）以上都不对

5. 在线索化二叉树中，t 所指结点没有左子树的充要条件是什么？

t->ltag == 1

数据结构与算法三

（1）图结构习题

1. 在一个图中，所有顶点的度数之和与所有边数之间存在什么关系？

一条边对应两个度，故所有顶点的度数之和等于总边数的两倍。

2. 在一个有向图中，所有顶点的入度之和与所有顶点的出度之和之间存在什么关系？

一条弧必有一个弧头和一个弧尾，弧头对应一个入度，弧尾对应一个出度，故所有顶点的入度之和等于出度之和。

3. 一个有 n 个顶点的无向图最多有多少条边？

任意两点之间可有一条边，所以最多有 $C_n^2 = \frac{n(n-1)}{2}$ 条边。

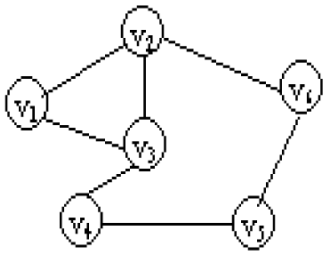
4. 具有 6 个顶点的无向图至少应有多少条边才能确保是一个连通图？

至少有 5 条边。

5. 对于一个具有 n 个顶点和 e 条边的无向图，若采用邻接表表示，则表头数组的大小是多少？所有邻接表中的结点总数是多少？

需要 n 个头结点和 2e 个表结点，则表头数组大小为 n，结点总数为 n+2e。

6. 根据右图，写出其邻接矩阵表达式；并从 v2 出发按深度优先搜索和广度优先搜索算法遍历得到所有可能的顶点序列。



邻接矩阵：

$$\begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{array} \begin{array}{c} \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

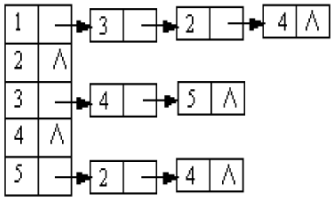
深度优先搜索：

$$V_2 \Rightarrow V_1 \Rightarrow V_3 \Rightarrow V_4 \Rightarrow V_5 \Rightarrow V_6$$
$$V_2 \Rightarrow V_3 \Rightarrow V_1 \Rightarrow V_4 \Rightarrow V_5 \Rightarrow V_6$$
$$V_2 \Rightarrow V_3 \Rightarrow V_4 \Rightarrow V_5 \Rightarrow V_6 \Rightarrow V_1$$
$$V_2 \Rightarrow V_6 \Rightarrow V_5 \Rightarrow V_4 \Rightarrow V_3 \Rightarrow V_1$$

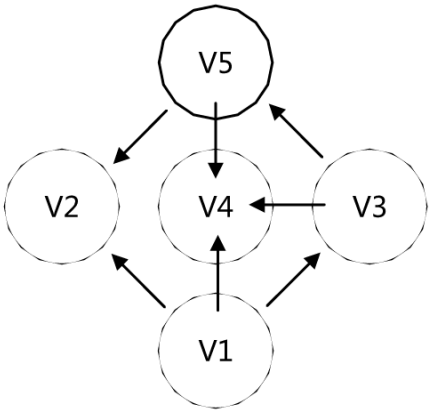
广度优先搜索：

$$V_2 \Rightarrow V_1 \Rightarrow V_3 \Rightarrow V_6 \Rightarrow V_4 \Rightarrow V_5$$
$$V_2 \Rightarrow V_1 \Rightarrow V_6 \Rightarrow V_3 \Rightarrow V_5 \Rightarrow V_4$$
$$V_2 \Rightarrow V_3 \Rightarrow V_1 \Rightarrow V_6 \Rightarrow V_4 \Rightarrow V_5$$
$$V_2 \Rightarrow V_3 \Rightarrow V_6 \Rightarrow V_1 \Rightarrow V_4 \Rightarrow V_5$$
$$V_2 \Rightarrow V_6 \Rightarrow V_1 \Rightarrow V_3 \Rightarrow V_5 \Rightarrow V_4$$
$$V_2 \Rightarrow V_6 \Rightarrow V_3 \Rightarrow V_1 \Rightarrow V_5 \Rightarrow V_4$$

7. 一个有向图的邻接表如右图所示，画出该图的逻辑结构，并求出根据深度优先搜索算法从顶点 v1 出发遍历得到的顶点序列。



逻辑结构：



得到的顶点序列：

$$V_1 \Rightarrow V_2 \Rightarrow V_3 \Rightarrow V_4 \Rightarrow V_5$$
$$V_1 \Rightarrow V_2 \Rightarrow V_3 \Rightarrow V_5 \Rightarrow V_4$$
$$V_1 \Rightarrow V_2 \Rightarrow V_4 \Rightarrow V_3 \Rightarrow V_5$$
$$V_1 \Rightarrow V_3 \Rightarrow V_4 \Rightarrow V_5 \Rightarrow V_2$$
$$V_1 \Rightarrow V_3 \Rightarrow V_5 \Rightarrow V_2 \Rightarrow V_4$$
$$V_1 \Rightarrow V_4 \Rightarrow V_2 \Rightarrow V_3 \Rightarrow V_5$$
$$V_1 \Rightarrow V_4 \Rightarrow V_3 \Rightarrow V_5 \Rightarrow V_2$$

（2）排序习题

1. 已知一个单链表由 3000 个整数元素组成，其值在 1 - 1000000 之间，分析上述排序方法中哪些可用于这个链表的排序？

理论上上述算法都可以实现，但是单链表的特点是顺序查找并进行操作，所以比较适用于单链表的排序方法有：直接插入排序，起泡排序，简单选择排序。

2. 分别使用折半插入排序和简单选择排序法写出对含有 4 个记录的序列进行排序过程，设四个记录均不相等。

设四个记录分别为 a1，a2，a3，a4，折半插入排序过程：

S1：	设 a1 为比较元素，比较 a1，a2 大小； 若 a1>a2，则排序为 a2，a1； 若 a1<a2，则排序为 a1，a2， 最终序列设为 b1，b2；
S2：	设 b1 为比较元素，比较 b1，a3 大小； 若 b1>a3，则排序为 a3，b1，b2； 若 b1<a3，则比较 b2，a3 大小； 若 b2>a3，则排序为 b1，a3，b2； 若 b2<a3，则排序为 b1，b2，a3； 最终序列设为 b1，b2，b3；
S3：	设 b2 为比较元素，比较 b2，a4 大小； 若 b2>a4，则比较 b1，a4 大小； 若 b1>a4，则排序为 a4，b1，b2，b3； 若 b1<a4，则排序为 b1，a4，b2，b3； 若 b2<a4，则比较 b3，a4 大小； 若 b3>a4，则排序为 b1，b2，a4，b3； 若 b3<a4，则排序为 b1，b2，b3，a4； 最终排序设为 b1，b2，b3，b4。

简单选择排序过程：

S1：	比较 a1，a2，a3，a4 大小，最小值设为 b1， 其余三个元素设为 a2，a3，a4；
S2：	比较 b2，b3，b4 大小，最小值设为 b2， 其余两个元素设为 a3，a4；
S3：	比较 a3，a4 大小，最小值设为设为 b3， 其余一个元素设为 a4；

S4： a4 设为 b4，最终排序为 b1，b2，b3，b4。

3. 用 C/C++语言写出以单链表为存储结构的简单选择排序算法。

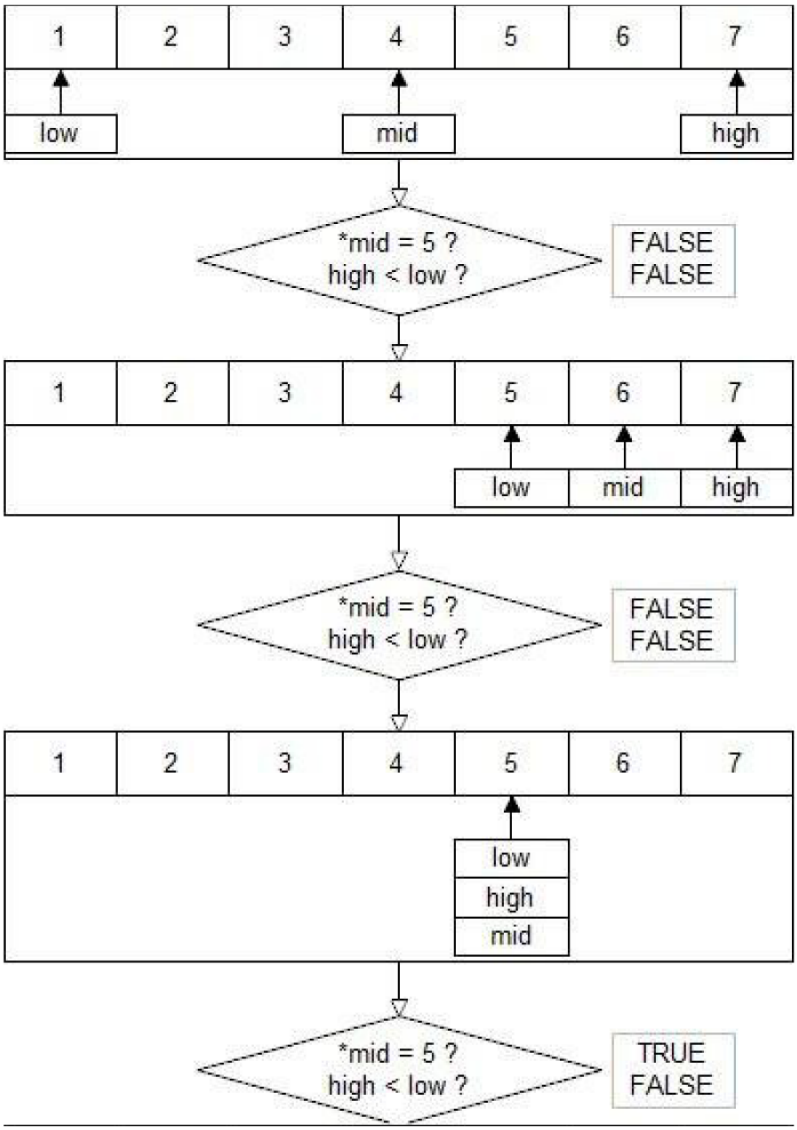
设给定的单链表以 head 为头指针 ,完成排序之后序列存储在以 L 为头指针的单链表中，简单选择排序算法如下：

```
int *p = L;
for(int *j = head; j != NULL; j = j->next, p = p->next)
{
    auto t = j->data;
    for(int *i = p; i != NULL; i = i->next)
    {
        if(i->data < t)
        t = i->data;
    }
    p->data = t;
}
```

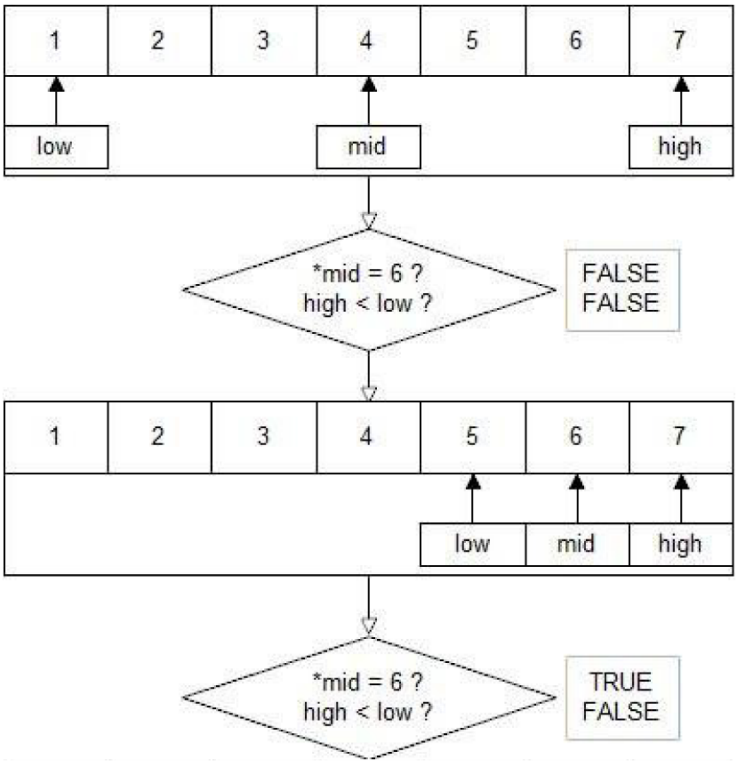
数据结构与算法四

1.分别画出在线性表中(1,2,3,4,5,6,7)中进行折半查找，获取关键字 5,6,7 的过程。

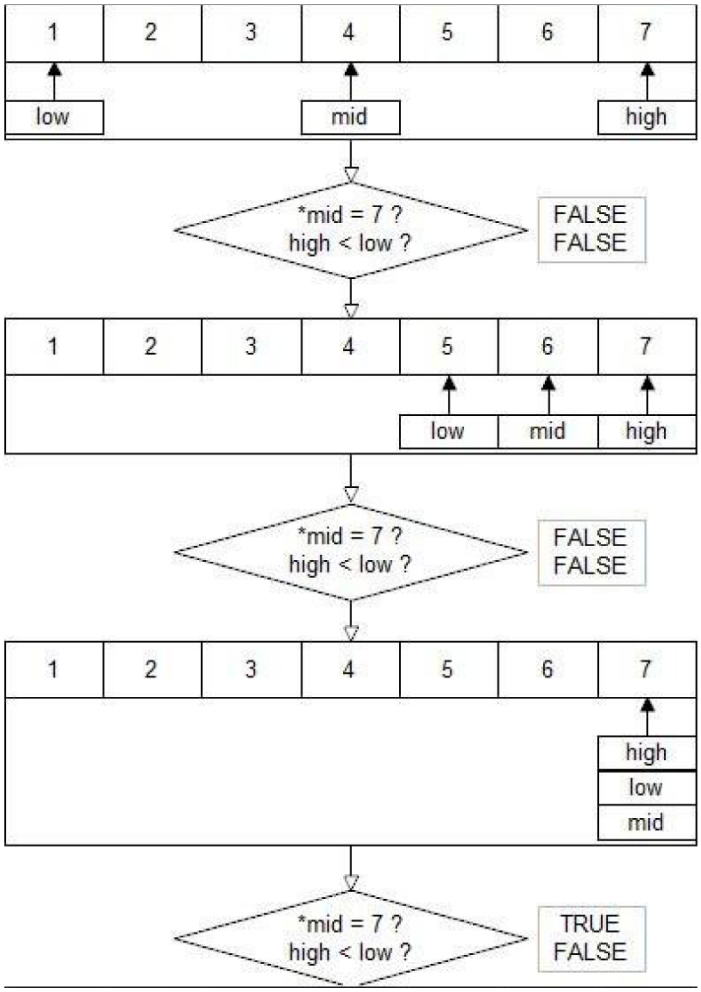
查找 5：



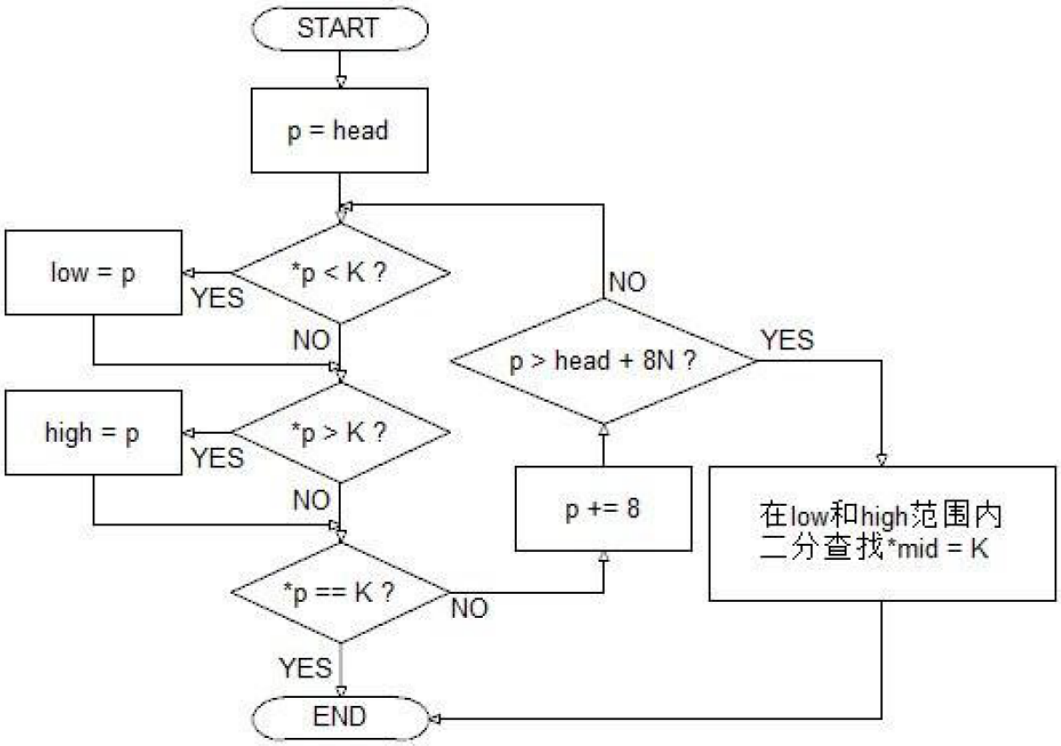
查找 6：


1
2
3
4
5
6
7
low
mid
high
*mid = 6 ?
high < low ?
FALSE
FALSE
1
2
3
4
5
6
7
low
mid
high
*mid = 6 ?
high < low ?
TRUE
FALSE

查找 7：


1
2
3
4
5
6
7
low
mid
high
*mid = 7 ?
high < low ?
FALSE
FALSE
1
2
3
4
5
6
7
low
mid
high
*mid = 7 ?
high < low ?
FALSE
FALSE
1
2
3
4
5
6
7
high
low
mid
*mid = 7 ?
high < low ?
TRUE
FALSE

2.已知一个有序表长为 8N，表中无关键字相同的记录。按照如下方法查找一个关键字等于给定值 K 的记录：先在第 8, 16, 24, …, 8K, …, 8N 个记录中进行顺序查找，如果查找成功则结束；否则确定一个折半查找的范围，继续查找，请以流程图形式画出这个过程。



3. 为下列关键字建立一个装载因子不小于 0.7 的哈希表（ZHAO, QIAN, SUN, LI, ZHOU, WU, ZHENG, WANG, FENG, CHEN, CHU,WE）。

构造哈希函数：

$$f\left(key\right)=[(Ord(Initial(Key))-Ord\left('A'\right)-1)/2]$$

用再哈希法处理冲突，构造产生冲突地址 $H(key)$ 的再哈希函数：

$$H(key)=H(key)+1$$

不断迭代直到寻找到空地址。哈希表如下：

0	1	2	3	4	5	6	7	8
CHENG	CHU	FENG			LI			QIAN
9	10	11	12	13	14	15	16	
SUN		WU	ZHAO	ZHOU	ZHENG	WANG	WE	

记录数 n=12，,表长 m=17，装载因子α=n/m=0.706。