一、选择题

1、设每个字符占一个字节，二维数组 A 的每个元素是由 5 个字符组成的串，其行下标从 0 到 7，其列下标从 0 到 9，若 A 按行优先存储，元素 A[6][4]的起始地址与当 A 按列优先存储时的哪个元素的起始地址相同？(　)

A、　A [2][8]　　B、　A[4][7]　　C、　A[6][7]　　D、A[0][8]

2、一棵含有 N (N>1)个结点的二叉树，在中序线索化前后，其空指针域个数分别是（　　）。

A、　N+1, 0　　B、　N+1, 2　　C、　N-1, 2　　D、N-1, 0

3、已知有向图 G=(V，E)，其中 V={a,b,c,d,e}，E={<a,b>，<a,c>，<d,c>，<d,e>，<b,e>，　<c,e>}，对该图进行拓扑排序，下面序列中不是拓扑序列的是（　　）。

A、　a,d,c,b,e　　B、　d,a,b,c,e　　C、　a,b,d,c,e　　D、　a,b,c,d,e

4、设 N 个待排序记录，初始状态已十分接近有序，用直接插入排序、直接选择排序和快速排序算法对其进行排序，这些算法的时间复杂度应为(　　)。

A、　$O(N),O(N^2),O(N)$　　B、　$O(N), O(N),O(N*\log_2 N)$

C、　$O(N),O(N^2),O(N^2)$　　D、　$O(N2), O(N),O(N*\log_2 N)$

二、简答题：

1、 已知二叉树的中根序列：BGDAECF 和后根序列：GDBEFCA.

（1）画出这棵二叉树；

（2）画出其自然对应的森林；

（3）写出其自然对应的森林的层次序列；

（4）画出定义在该二叉树上的中序线索二叉树。

2、 试画出下列并查集（Union/Find Set）操作序列的每步的结果：Union(1,2),Union(3,4),Union(2,4),Union(5,6),Find(2),Union(3,5),Find(4)。并查集初始规模为 6. Union(i,j)操作使用了按秩合并策略；Find 操作返回集合的代表元，使用了路径压缩策略。

3、 画出对下列整数序列进行基数排序的全过程（178，209，94，303，56，259，985，9，871，33）。

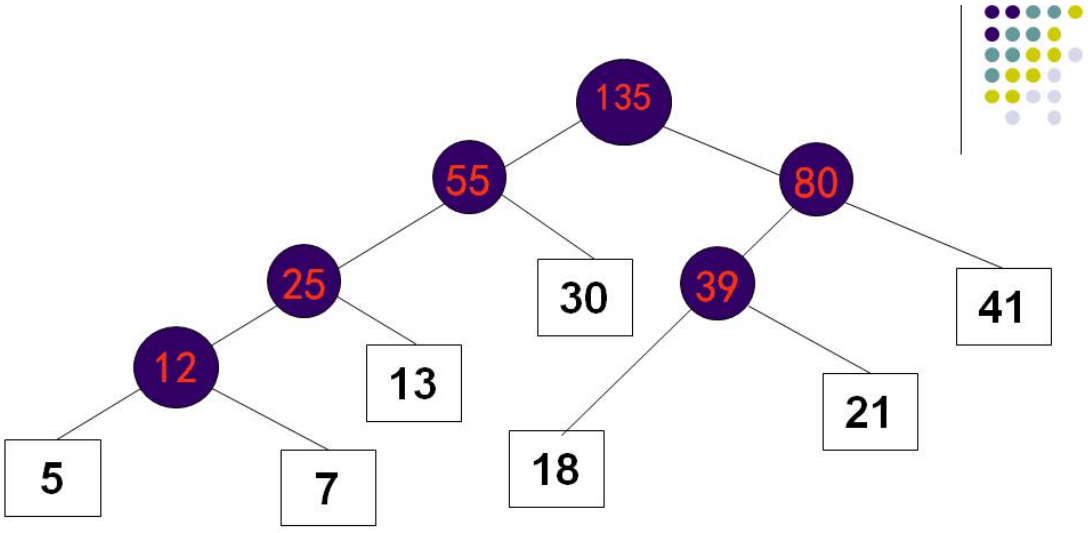
4、什么样的图其最小支撑树是唯一的？请构造一个这样的图，并给出其最小支撑树。

5、Prim 算法和 Kruskar 算法的时间复杂性是多少？分别适合求解哪类图？

6、T 是一颗非空二叉树。令 a 是 T 中空子树的个数，b 是 T 中节点个数。问：a 和 b 有什么关系？

7、观察后序线索二叉树，有几个空指针，并给出解释。

8、构造权值为 { 5, 13, 21, 7, 18, 30, 41 }的哈夫曼树。



{ 5, 13, 21, 7, 18, 30, 41 }

9、已知下列关键词和它们对应的散列函数值：

key	Zhao	Sun	Li	Wang	Chen	Liu	Zhang
H(key)	6	5	7	4	1	6	4

由此构造哈希表，用线性探查法解决冲突，计算平均查找长度 ASL。若用拉链法解决冲突情况又如何？

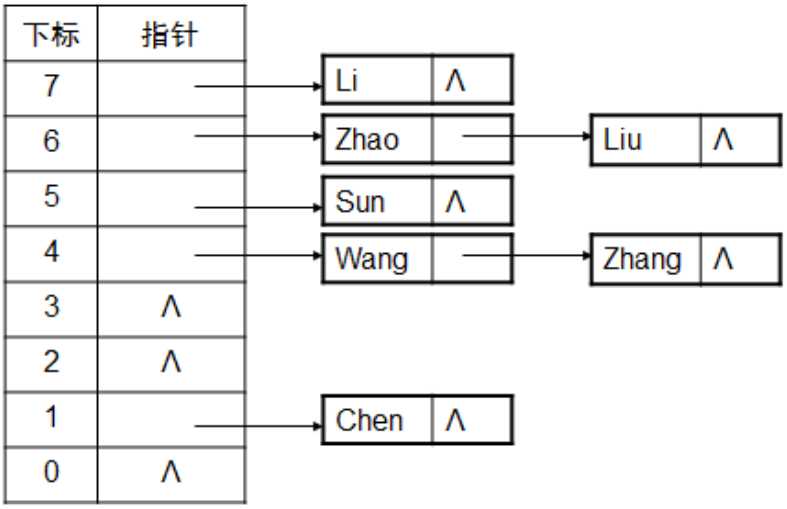
答案：

- 线性探测法（假设散列表的长度是8，下标从0开始）

下标	元素
7	Li
6	Zhao
5	Sun
4	Wang
3	
2	Zhang
1	Chen
0	Liu

- 查找成功ASL
  =(1*5+3*1+7*1)/7
  =15/7

- 拉链法（假设散列表长度是8，下标从0开始）



- 查找成功ASL=(1*5+2*1+2*1)/7=9/7

10、已知序列（17，31，13，11，20，35，25，8，4，11，24，40，27），请画出该序列的二叉查找树，并分别给出下列操作后的二叉树：

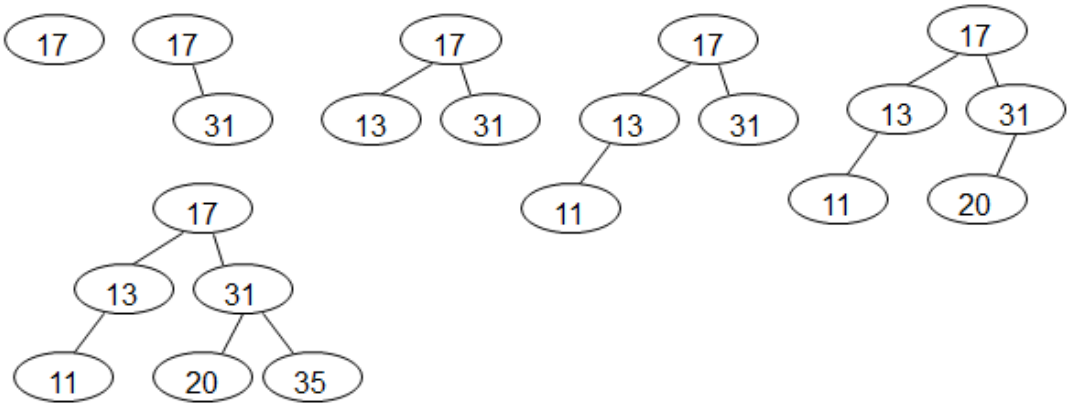
1）插入数据 9；

2）删除结点 17；

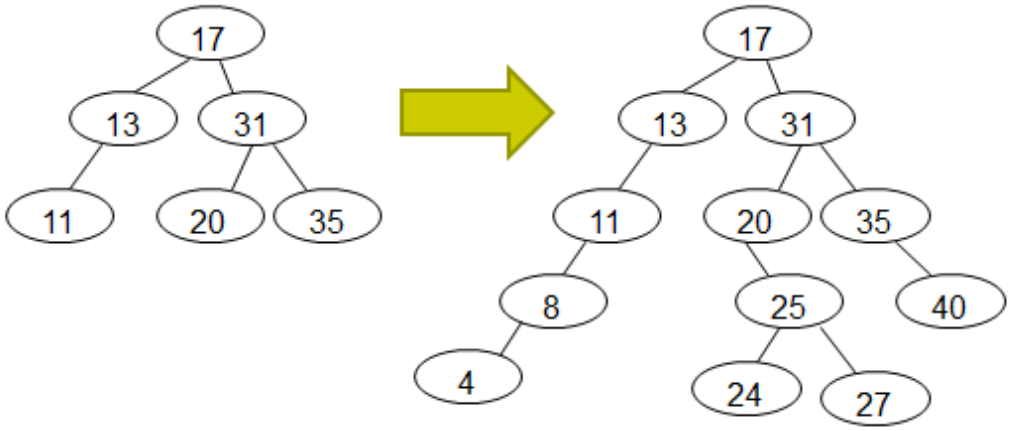
3）再删除结点 13.

答案：

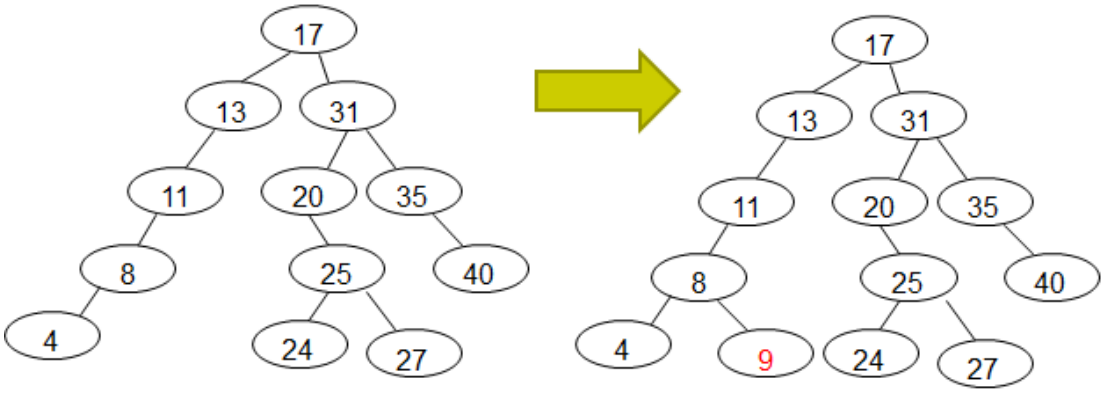
- 动态插入创建二叉查找树：
  (17,31,13,11,20,35,25,8,4,11,24,40,27)



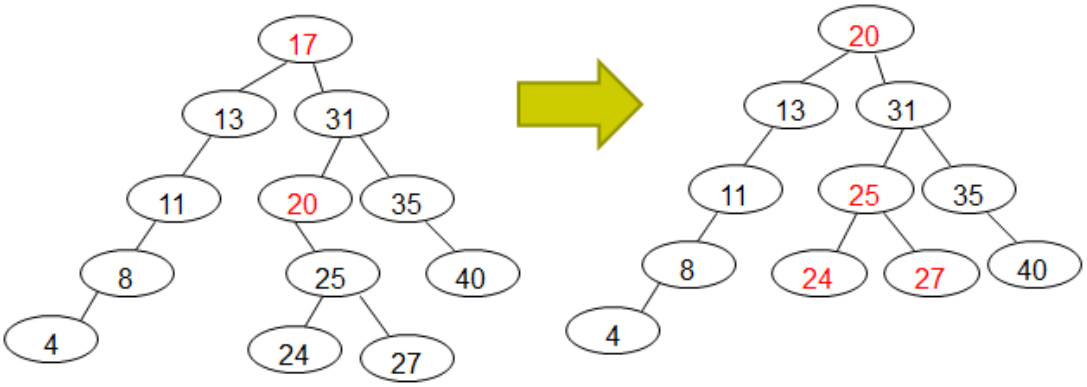
- (17,31,13,11,20,35,25,8,4,11,24,40,27)



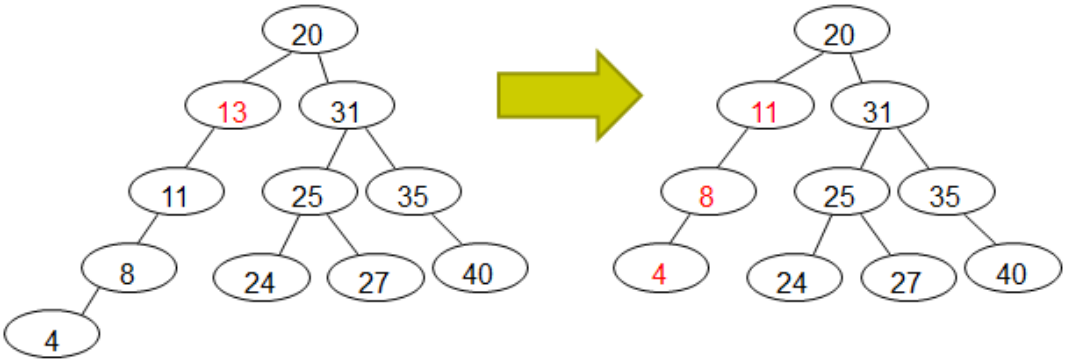
- 插入数据9



- 删除17



- 再删除节点13



11、k 阶斐波那契序列定义如下：

$$f_n = \begin{cases} 0 & ,0 \le n < k-1 \\ 1 & ,n = k-1 \\ f_{n-1} + f_{n-2} + \cdots f_{n-k} & ,n > k-1 \end{cases}$$ ，n 是非负整数

试计算 1 阶斐波那契序列的 f_{100} 和 5 阶斐波那契序列的 f_8 。

12、以 S 和 X 分别表示入栈和出栈操作，由 S 和 X 组成的序列称为合法序列当且仅当：当栈初始状态为空，该序列每一个操作都能被正确执行且栈的终态为栈空（空栈的出栈操作无法正确执行；假设栈无限大，所有入栈都可以执行）。例如： SXSX 为合法序列， SXXS 为非法序列。

1) 试给出区分给定序列为合法序列或非法序列的一般准则；

2) 两个不同的合法操作序列不可能得到相同的输出序列。这一判断是否正确？给出判断依据（证明或反例）。

13、已知主串 s ＝ "abcaabbabcabaacbacba"，模式串 p ＝ "abcabaa"，写出模式串的失败函数 f 值，并由此画出 KMP 算法匹配的全过程。

14、一个矩阵从 A[0][0]开始存放，每个元素占 4 个存储单元，若 A[7][8]的存储地址为 2732，A[13][16]的存储地址为 3364。关于此矩阵存储的方式下列哪个叙述是正确的，并给出具体的分析过程。

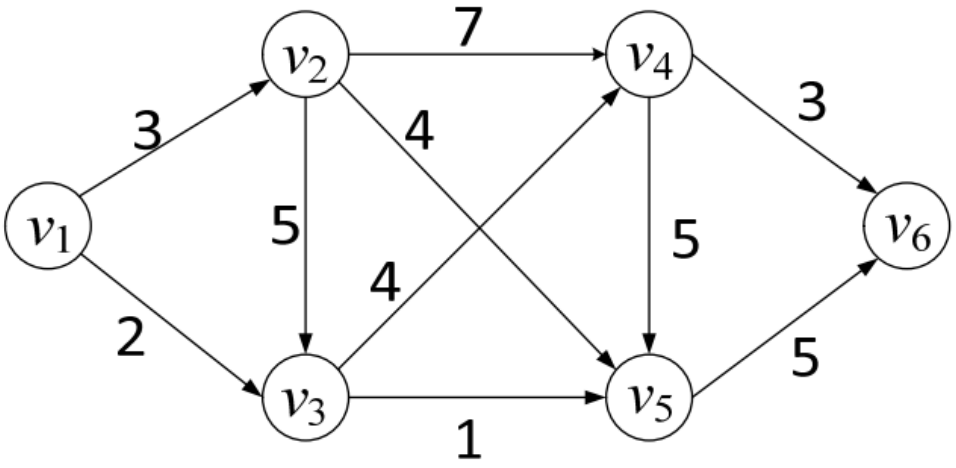
[A] 只能按行优先存储

[B] 只能按列优先存储

[C] 按行优先存储或按列优先存储均可

[D] 因计算有误，按[A]、 [B]、 [C]三种存储方式都不行

15、对于如下有向图，求顶点 v1 到顶点 v6 的最短路径，第二短路径，以及第三短路径，要求写出求解的具体过程。



16、

设有五个数据 do，for，if,repeat，while，它们排在一个有序表中，其查找概率分别为 p_1 =0.2, p_2=0.15，p_3=0.1，p4=0.03，p_5=0.01。而查找它们之间不存在数据的概率分别为 q0=0.2,q1=0.15,q2=0.1,q3=0.03,q4=0.02,q5=0.01。

do		for		if		repeat		while		
q_0	p_1	q_1	p_2	q_2	p_3	q_3	p4	q_4	p_5	q_5

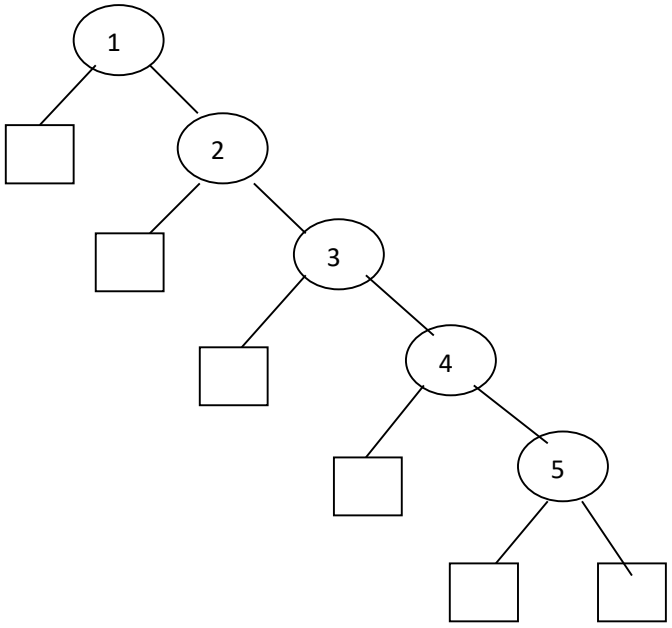
(1) 试画出对该有序表采用顺序查找时的判定树和采用折半查找时的判定树。

(2) 分别计算顺序查找时的查找成功和不成功的平均查找长度，以及折半查找时的查找成功和不成功的平均查找长度。

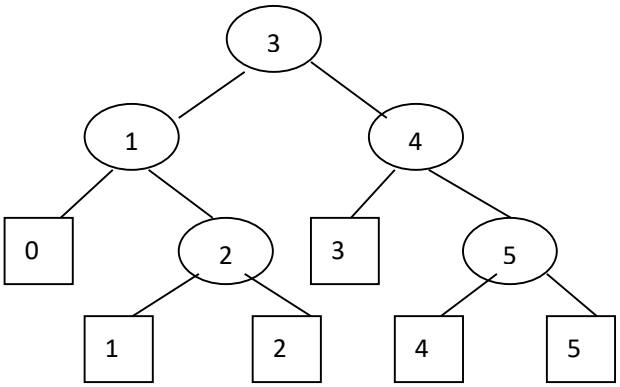
(3) 判定是顺序查找好？还是折半查找好？

答：

顺序查找时的判定树



采用折半查找时的判定树



(2) ASL $_{\text{顺序成功}}$ =（$1p_1+2p_2+3p_3+4p_4+5p_5$)=0.97

ASL $_{\text{折半成功}}$ =（$1p_3+2（p_1+p_4）+3（p_2+p_5$)=1.04

ASL $_{\text{折半失败}}$ =（$2q_0+3q_1+3q_2+2q_3+3q_4+3q_5$)=1.30

ASL $_{\text{顺序失败}}$ =（$1q_0+2q_1+3q_2+4q_3+5q_4+5q_5$)=1.07

（3）本题中顺序检索好。

17、

设有序顺序表为｛10, 20, 30, 40, 50, 60, 70, 80}，采用折半查找时，查找成功的平均查找长度是多少？

【解体思路】画出相应的二叉判定树，计算查找成功的平均查找次数（查找长度）。

【解答】该题是关于折半查找的，包含 8 个元素。相应的二叉判定树如下图 9.2：

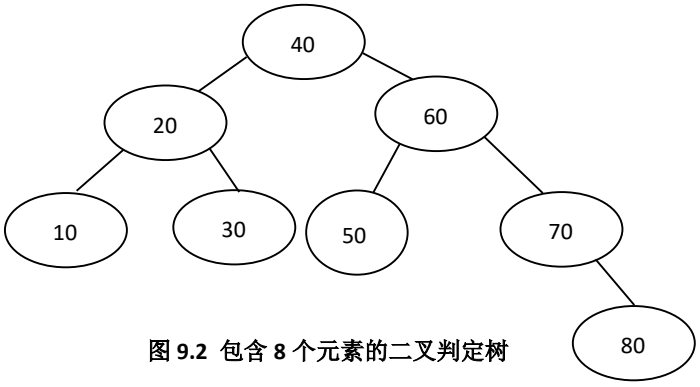


图 9.2 包含 8 个元素的二叉判定树

很明显，查找成功的平均比较次数是 $(1□2□2□3□3□3□3□4)\big/_{8}$ □21/8，查找失败的平均比较次数是 $(3□3□3□3□3□3□4□4)\big/_{9}$ □29/9 。

18、将(for, case, while, class, protected, virtual, public, private, do, template, const ,if, int)中的关键字依次插入初态为空的二叉查找树中，请画出所得到的树 T。然后画出删去 for 之后的二叉查找树 T'，若再将 for 插入 T'中得到的二叉查找树 T"是否与 T 相同?最后给出 T"的先序、中序和后序序列。

【解体思路】 按照二叉查找树的建造算法生成一棵二叉查找树，并且根据插入和删除算
法进行操作。

【解答】 将所有关键字插入后得到的树型 T 如下：

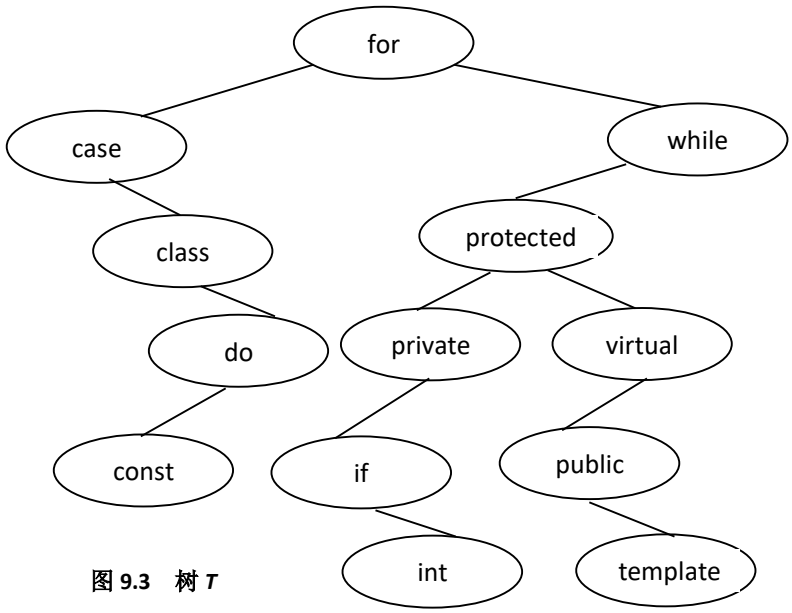


图 9.3　树 *T*

删除关键字 for 后，得到的树型 T' 如下：

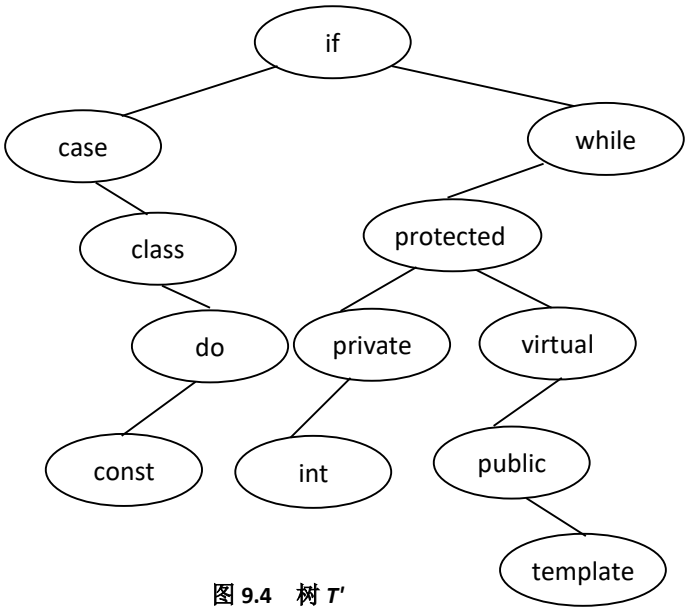


图 9.4　树 *T'*

重新插入关键字 for 之后，得到的树型 T'' 如下：

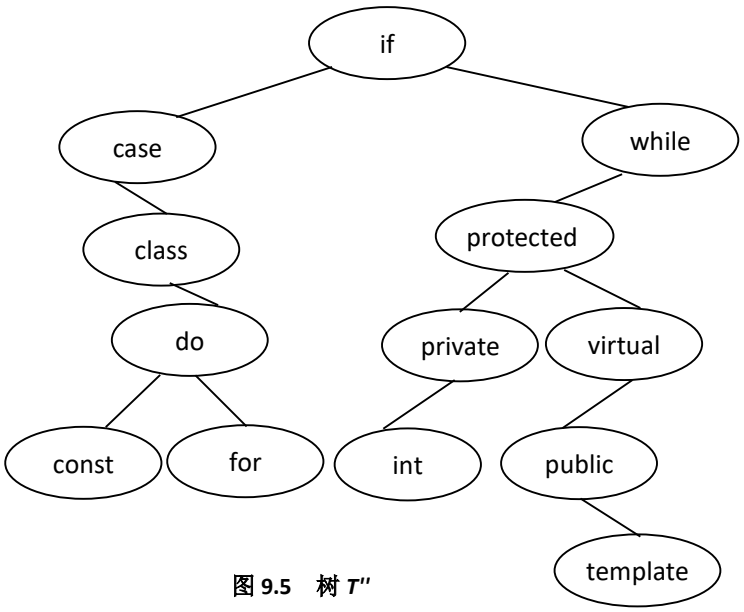


图 9.5　树 T''

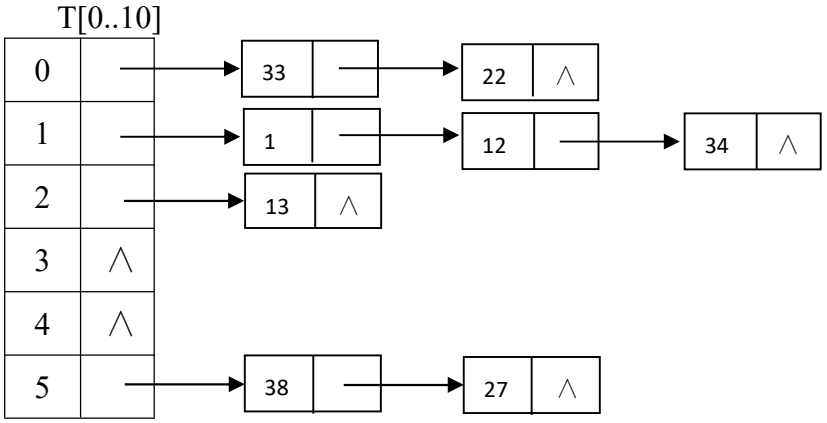
很显然，重新插入后的树型与原来的树型并不相同，这也从另一方面揭示了关键字输入的顺序不同，会导致不同形状的二叉查找树。树型 T''的先序是 if, case, class, do, const, for, while, protected, private, int, virtual, public, template；中序是 case, class, const, do, for, if, int, private, protected, public, template, virtual, while ；后序是 const, for, do, class, case, int, private, template, public, virtual, protected, while, if ．

19、

设散列表长度为 11，散列函数 $h(x)$ □ x%11，给定的关键字序列为：1，13，12，34，38，33，27，22. 试画出分别用拉链法和线性探查法解决冲突时所构造的散列表，并求出在等概率情况下，这两种方法查找成功和失败时的平均查找长度。

【解体思路】根据散列函数以及解决冲突的两种方法，构造出相应的散列表，再进行分析。

【解答】拉链法如图



6	∧
7	∧
8	∧
9	∧
10	∧

拉链法

线性探查法如图 9.7：

下标	0	1	2	3	4	5	6	7	8	9	10
T[0..10]	33	1	13	12	34	38	27	22			
探查次数	1	1	1	3	4	1	2	8			

线性探查法

用拉链法查找成功平均查找长度为：

ASL_{suss} □(1*4+2*3+3*1)/8=1.625

查找失败时平均查找长度为：

ASL_{unsucc} □(2+3+1+0+0+0+2+0+0+0+0)/11=0.73

用线性探查法查找成功时平均查找长度为：

ASL_{succ} □(1+1+1+3+4+1+2+8)/8=2.625

查找失败时平均查找长度为：

ASL_{unsucc} □(9+8+7+6+5+4+3+2+1+1+1)/11=4.3

三、填空题

1、

```
算法 PartA（R, s, e）// 分划文件（R_s, R_{s+1}, …, R_e），且 K_{s-1}=-∞， K_{e+1}=+∞
PA1 [初始化]
    i←s . j←[①]
    K←K_s . R←R_s .
PA2 [分划过程]
    WHILE i<j DO
    （j←j-1 . 
      WHILE [②] DO j←j-1 . 
      IF i≥j THEN j←i
      ELSE
          （R_i←R_j .
           i←i+1 . 
           WHILE K_i<K DO i←i+1 .
           IF [③] THEN R_j←R_i ) ) .
PA3 [④] ▌
```

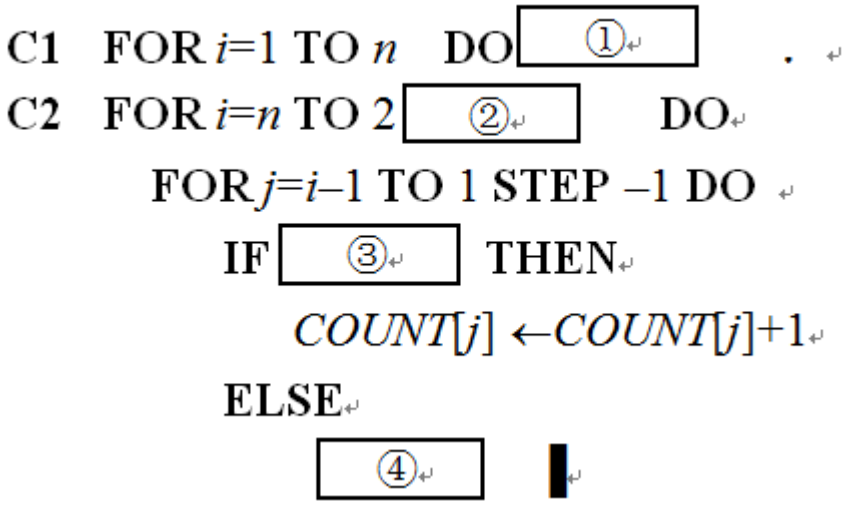
正确答案：

1.e+1　2.K_j>=K　3.i<j　4.R_j<-R

2、算法 C(R，n)

// 比较计数，本算法按关键词 K1，K2，…，Kn 排序记录 R1，R2，…，Rn .

// 一维数组 COUNT[1 : n]用来记录各个记录的排序位置



正确答案：

1. count[i]<-1 2. STEP-1 3. Kj>K 4. count[i]<-count[i]+1

四、算法题

1、设计一算法，在尽可能少的时间内重排数组，使所有取负值的关键词放在所有取非负值的关键词之前，并分析算法的时间复杂度。

答案：

```
算法Part(A,n.A)
/*以0为基准元素一次分划*/
P1[ 初始化]
        i←1 .  j←n.
P2[以0分划]
        WHILE  i<j  DO
            (
               WHILE  K_i<0 AND i<j DO  i←i+1 .
               WHILE  K_j>0  AND i<j DO  j←j–1.
               IF  i<j  THEN  R_i ↔ R_j .  )
```

2、奇偶交换排序算法的基本思想描述如下：排序过程通过对文件 $x[i]$（$1\le i\le n$）的若干次扫描来完成，第奇数次扫描，对所有下标为奇数的记录 $x[i]$与其后面的记录 $x[i+1]$（$1\le i\le n-1$）相比较，若 $x[i].KEY$（记录 $x[i]$的关键词）＞ $x[i+1].KEY$，则交换 $x[i]$和 $x[i+1]$的内容；第偶数次扫描，对所有下标为偶数的记录 $x[i]$与其后面的记录 $x[i+1]$（$2\le i\le n-1$）相比较，若 $x[i].KEY>x[i+1].KEY$，则交换 $x[i]$和 $x[i+1]$之内容，重复上述过程直到排序完成为止。

（1）排序的终止条件是什么？

（2）完成该算法的具体设计。

- 算法Sort(X,n)
- S1[一趟扫描过程中，均没有记录交换则算法终止]

```
change ← 1.
while (change)
 (change ← 0.
   for i ← 1 to n-1 step 2  //奇交换
     if (X[i].key>X[i+1].key)
       ( X[i] ↔ X[i+1].
          change ← 1.)
  for i ← 2 to n-1 step 2  //偶交换
    if (X[i].key>X[i+1].key)
       ( X[i] ↔ X[i+1].
          change ← 1.)
 ))
```

3、 按照对满二叉树编号的方法，对一棵高度为 h 的满 k 叉树进行编号，次序为从上到下，同层上从左到右，编号从 0 开始。

（1）树的第 i (0< i < h)层上有多少个结点？

（2）树的第 h 层上可能有多少个结点？

4、 编写算法判断两棵二叉树 T 和 T’是否相似。两棵二叉树相似是指它们具有相同结构。

```
算法Like(t1,t2)
/*判断两棵二叉树是否相似,t1,t2表示两棵树的根节点。若相似，返回值为true
    ，否则为false*/
L1[递归出口]
   IF  t1=NULL AND t2=NULL THEN RETURN true.
   IF  t1=NULL OR t2=NULL THEN RETURN false.
L2[递归调用]
   RETURN Like(left(t1),left(t2)) AND Like(right(t1),right(t2)). ▌
时间复杂度为O(n1+n2)
```

5、编写算法计算二叉树中边的个数。

```
算法E(t.n)
/*计算二叉树t的边数，结果放在n中*/
L1[递归出口]
   n ←0.
   IF  t=NULL THEN RETURN.
L2[递归调用]
   IF (left(t)<>NULL) THEN(E(left(t),n1).n ←n+1+n1.).
   IF (right(t)<>NULL) THEN(E(right(t),n2).n ←n+1+n2.).
```

6、编写判定给定的二叉树是否是二叉查找树的算法。

【提示】判定二叉树是否为二叉查找树同样是建立在中根遍历的框架基础下，在遍历中附设一指针 pre 指向当前访问结点的中根直接前驱，每访问一个结点便比较前驱结点 pre 和此结点是否有序，若遍历结束后各结点和其中根直接前驱均满足有序，则此二叉树为二叉查找树；否则，只要有一个结点不满足，那么此二叉树就不是二叉查找树。

答案：

```
算法BST (t.pre,flag)
/* 使用中根遍历和pre指针判断t是否是二叉查找树*/
B1 [特判]
    flag ← TRUE.
    IF t = NULL THEN RETURN.
B2 [中根遍历]
     BST(left(t),pre,flag).
     IF(!flag) THEN RETURN.
     IF( pre!=NULL   AND   data(pre)>data(t)  )
            THEN(flag ← FALSE. RETURN.).
      pre ←t.
     BST(right(t),pre,flag). ▌
```