



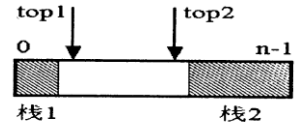


- A. (5, 1, 4, 3, 6, 2, 8, 7)      B. (5, 1, 4, 3, 2, 6, 7, 8)  
 C. (5, 1, 4, 3, 2, 6, 8, 7)      D. (8, 7, 6, 5, 4, 3, 2, 1)
14. 分块查找方法将表分为多块, 并要求 (      )  
 A. 块内有序      B. 块间有序  
 C. 各块等长      D. 链式存储
15. 便于进行布尔查询的文件组织方式是 (      )  
 A. 顺序文件      B. 索引文件  
 C. 散列文件      D. 多关键字文件

**二、填空题 (本大题共 10 小题, 每小题 2 分, 若有两个空格, 每个空格 1 分, 共 20 分)**

请在每个空格中填上正确答案。错填、不填均无分。

16. 数据的链式存储结构的特点是借助\_\_\_\_\_表示数据元素之间的逻辑关系。  
 17. 如果需要对线性表频繁进行\_\_\_\_\_或\_\_\_\_\_操作, 则不宜采用顺序存储结构。  
 18. 如图所示, 可以利用一个向量空间同时实现两个类型相同的栈。其中栈 1 为空的条件是  $top1=0$ , 栈 2 为空的条件是  $top2=n-1$ , 则“栈满”的判定条件是\_\_\_\_\_。
19. 静态存储分配的顺序串在进行插入、置换和\_\_\_\_\_等操作时可能发生越界。  
 20. 广义表  $L = (a, (b, ()))$  的深度为\_\_\_\_\_。  
 21. 任意一棵完全二叉树中, 度为 1 的结点数最多为\_\_\_\_\_。  
 22. 求最小生成树的克鲁斯卡尔(Kruskal)算法耗用的时间与图中\_\_\_\_\_的数目正相关。  
 23. 在 5 阶 B-树中, 每个结点至多含 4 个关键字, 除根结点之外, 其他结点至少含\_\_\_\_\_个关键字。  
 24. 若序列中关键字相同的记录在排序前后的相对次序不变, 则称该排序算法是\_\_\_\_\_的。  
 25. 常用的索引顺序文件是\_\_\_\_\_文件和\_\_\_\_\_文件。



题 18 图

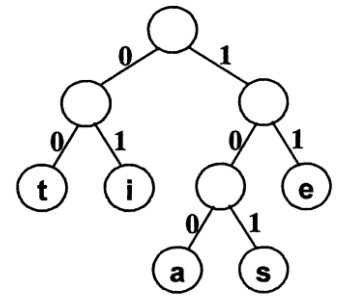
**三、解答题 (本大题共 4 小题, 每小题 5 分, 共 20 分)**

26. 如图所示, 在  $n \times n$  矩阵 A 中, 所有下标值满足关系式  $i+j < n+1$  的元素  $a_{ij}$  的值均为 0, 现将 A 中其它元素按行优先顺序依次存储到长度为  $n(n+1)/2$  的一维数组 sa 中, 其中元素  $a_{1,n}$  存储在 sa [0]。  
 (1) 设  $n=10$ , 元素  $a_{4,9}$  存储在 sa [p], 写出下标 p 的值;  
 (2) 设元素  $a_{i,j}$  存储在 sa [k] 中, 写出由 i,j 和 n 计算 k 的一般公式。

$$\begin{bmatrix} 0 & \cdots & \cdots & 0 & a_{1,n} \\ 0 & \cdots & 0 & a_{2,n-1} & a_{2,n} \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & a_{n-1,2} & \cdots & \cdots & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n} \end{bmatrix}$$

题 26 图

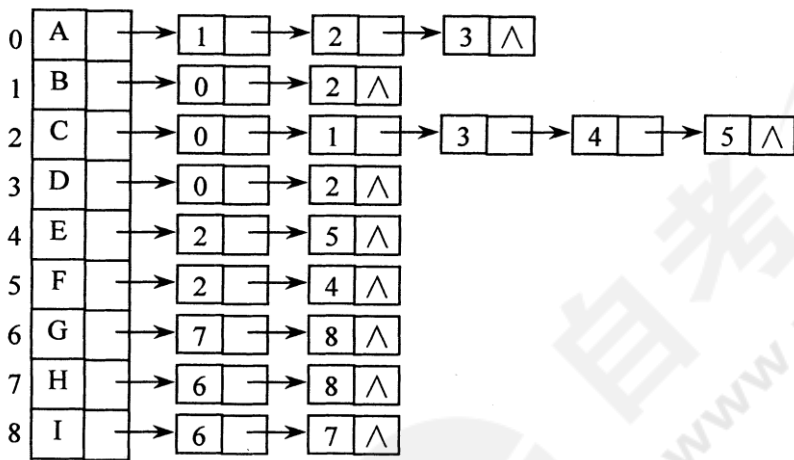
27. 由字符集 {s, t, a, e, I} 及其在电文中出现的频度构建的哈夫曼树如图所示。已知某段电文的哈夫曼编码为 111000010100, 请根据该哈夫曼树进行译码, 写出原来的电文。



题 27 图

28. 已知无向图 G 的邻接表如图所示，

- (1)画出该无向图；
- (2)画出该图的广度优先生成森林。



题 28 图

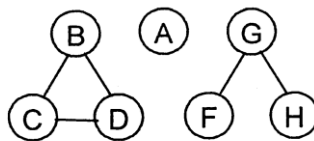
29. 对序列(48, 37, 63, 96, 22, 31, 50, 55, 11)进行升序的堆排序，写出构建的初始(大根)堆及前两趟重建堆之后的序列状态。

- 初始堆：  
第 1 趟：  
第 2 趟：

**四、算法阅读题(本大题共 4 小题，每小题 5 分，共 20 分)**

30. 阅读下列算法，并回答问题：

- (1)无向图 G 如图所示，写出算法 f30(&G)的返回值；



题 30 图

- (2)简述算法 f30 的功能。

```
#define MaxNum 20
int visited[MaxNum];
void DFS(Graph *g, int i);
/*从顶点 vi 出发进行深度优先搜索，访问顶点 vj 时置 visited[j]为 1*/
int f30(Graph *g)
```

```

{ int i, k;
  for (i=0; i<g->n; i++) /*g->n 为图 g 的顶点数目*/
    visited[i]=0;
  for (i=k=0; i<g->n; i++)
    if (visited[i]==0)
    { k++;
      DFS(g, i);
    }
  return k;
}
    
```

31. 假设学生成绩按学号增序存储在带头结点的单链表中，类型定义如下：

```

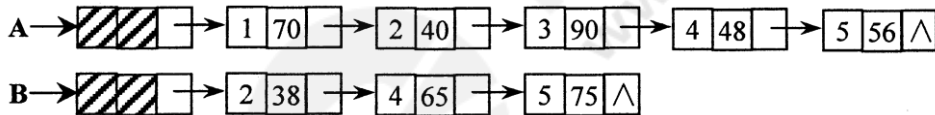
typedef struct Node {
    int id;           /*学号*/
    int score;       /*成绩*/
    struct Node *next;
} LNode, *LinkList;
    
```

阅读算法 f31，并回答问题：

(1) 设结点结构为 

id	score	next
----	-------	------

，成绩链表 A 和 B 如图所示，画出执行算法 f31(A, B) 后 A 所指的链表；



题 31 图

(2) 简述算法 f31 的功能。

```

void f31(LinkList A, LinkList B)
{ LinkList p, q;
  p=A->next;
  q=B->next;
  while (p && q)
  { if (p->id<q->id)
    { p=p->next;
      else if (p->id>q->id)
        q=q->next;
      else
        { if (p->score<60)
          if (q->score<60)
            p->score=q->score;
          else p->score=60;
          p=p->next;
          q=q->next;
        }
    }
  }
}
    
```

```

    }
}
}

```

32. 阅读下列算法，并回答问题：

- (1) 设串 s = "OneWorldOneDream", t = " One "，pos 是一维整型数组，写出算法 f32(s, t, pos) 执行之后得到的返回值和 pos 中的值；  
 (2) 简述算法 f32 的功能。

```

int  strlen(char*s);    /*返回串 s 的长度*/
int  index(char*st, char*t);
    /*若串 t 在串 st 中出现，则返回在串 st 中首次出现的下标值，否则返回-1*/
int  f32(char*s, char*t, int pos [ ] )
{   int  i, j, k, ls, lt;
    ls=strlen(s);
    lt=strlen(t);
    if (ls==0||lt==0) return-1;
    k=0;
    i=0;
    do {
        j=index(s+i, t);
        if (j>=0)
            {   pos [k++] =i+j;
                i+=j+1t;
            }
    }while(i+1t<=1s && j >=0);
    return  k;
}

```

33. 二叉排序树的存储结构定义为以下类型：

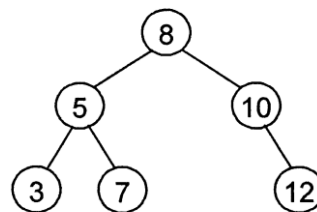
```

typedef int  KeyType;
typedef struct  node  {
    KeyType  key;           /*关键字项*/
    InfoType  otherinfo;   /*其它数据项*/
    struct  node  *lchild, *rchild; /*左、右孩子指针*/
}  BSTNode, *BSTree;

```

阅读算法 f33，并回答问题：

- (1) 对如图所示的二叉排序树 T，写出 f33(T, 8) 返回的指针所指结点的关键字；  
 (2) 在哪些情况下算法 f33 返回空指针？  
 (3) 简述算法 f33 的功能。



题 33 图

```

BSTNode  *f33(BSTree T, KeyType x)
{   BSTNode  *p;
    if (T==NULL) return  NULL;
    p=f33(T->lchild, x);
    if (p!=NULL)return p;
}

```

```
if (T->key>x)return T;  
return f33(T->rchild, x);  
}
```

### 五、算法设计题（本题 10 分）

34. 假设线性表采用顺序存储结构，其类型定义如下：

```
#define ListSize 100  
typedef struct {  
    int data [ListSize] ;  
    int length;  
} SeqList, *Table;
```

编写算法，将顺序表 L 中所有值为奇数的元素调整到表的前端。