

西南交大计算机、软工考研全套视频和资料，真题、考点、命题规律独家视频讲解  
详见：网学天地 (www.e-studysky.com)；咨询QQ: 3505993547

### 一、填空题 (1×20 分)

1. 8 层完全二叉树至少有\_\_\_\_\_个结点；拥有 100 个结点的完全二叉树其最大层号为\_\_\_\_\_。
2. 基于关键字比较大小的排序算法中，\_\_\_\_\_排序算法的平均时间复杂度最优。
3.  $n$  个顶点的有向图，至少需要\_\_\_\_\_条弧才能保证是连通图。
4. 对数组存储线性表 (16, 15, 32, 11, 6, 30) 用快速排序算法进行由小到大排序，若排序下标范围为 0~5，选择元素 16 作为支点，调用一趟快速排序算法后，元素 16 在数组中的下标位置为\_\_\_\_\_。
5. 用 S 表示入栈操作，X 表示出栈操作，若元素入栈顺序为 1234，为了得到 1342 出栈顺序，相应的 S、X 操作串为\_\_\_\_\_。
6. 中缀式  $a + b * 3 + 4 * (c - d)$  对应的前缀式为\_\_\_\_\_，若  $a=1, b=2, c=3, d=4$  则后缀式  $db/cc*a-b*+$  的运算结果是\_\_\_\_\_。
7. 用下标从 0 开始的  $N$  元数组实现循环队列时，为实现下标变量  $m$  加 1 后在数组有效下标范围内循环，可采用的表达式是  $m :=$ \_\_\_\_\_。

\_\_\_\_\_ (填 PASCAL, C 语言考生不填)；m =  
\_\_\_\_\_ (填 C 语言, PASCAL 考生不填)。

8. 在一棵含有 n 个顶点的非平衡二叉排序树中进行查找，平均时间复杂度的上限（即最坏情况平均时间复杂度）为\_\_\_\_\_。
9. 以下程序的功能是实现带附加头结点单链表数据结点逆序连接，请填空完善之。

{ PASCAL 语言, C 语言考生不答 }

PROCEDURE reverse (h: pointer);

{ h 为附加头结点指针; 类型 pointer 同算法设计第 3 题 }

VAR p, q: pointer;

BEGIN

p := h^.next; h^.next := NIL;

WHILE \_\_\_\_\_ DO

BEGIN

q := p; p := p^.next;

q^.next := h^.next; h^.next := \_\_\_\_\_

END

END;

/\* C 语言, PASCAL 语言考生不答 \*/

void reverse (pointer h)

/\* h 为附加头结点指针; 类型 pointer 同算法设计第 3 题 \*/

{ pointer p, q;

p = h->next; h->next = NULL;

while ( \_\_\_\_\_ )

{ q = p; p = p->next;

q->next = h->next; h->next = \_\_\_\_\_ ;

}

}

10. 以下程序是二叉链表树中序遍历的非递归算法，请填空使之完善。

二叉链表树结点结构类型定义如下：

TYPE bitree = ^ node; { PASCAL 语言 }

node = RECORD data: char; lchild, rchild: bitree END;

```
typedef struct node    /* C语言 */
```

```
    { char data ; struct node  *lchild , *rchild ;  
    } *bitree ;
```

```
{ PASCAL 语言, C 语言考生不答 }
```

```
PROCEDURE vst (bt : bitree); { bt 为根结点指针 ;
```

```
VAR p : bitree ;
```

```
BEGIN
```

```
    p := bt ;
```

```
    InitStack (s); { 初始化栈 s 为空栈 }
```

```
    WHILE (p <> NIL) OR (NOT Empty(s)) { 栈 s 不空 } DO
```

```
        IF p <> NIL
```

```
            THEN
```

```
                BEGIN
```

```
                    Push (s, p); { p 入栈 }
```

```
                END
```

```
            ELSE
```

```
                BEGIN
```

```
                    p := Pop (s); { 栈顶元素出栈 }
```

```
                    write (p^.data);
```

```
                END ;
```

```
    END ;
```

```
/* C 语言, PASCAL 语言考生不答 */
```

```
void vst (bitree bt) /* bt 为根结点指针 */
```

```
{ bitree p ;
```

```
    p = bt ;
```

```
    InitStack (s); /* 初始化栈 s 为空栈 */
```

```
    while (p || ! Empty(s)) /* 栈 s 不空 */
```

```
        if (p)
```

```
            { Push (s, p); /* p 入栈 */
```

```
            }
```

```
        else
```

```
{ p = Pop(s); /* 栈顶元素出栈 */  
  printf("%c", p->data);  
  _____  
}
```

11. 二叉树存储结构同上题，以下程序为求二叉树深度的递归算法，请填空完善之。

{ PASCAL 语言, C 语言考生不答 }

FUNCTION depth(bt: bitree): integer; { bt 为根结点指针 }

VAR hl, hr: integer;

BEGIN

IF bt = NIL

THEN depth := \_\_\_\_\_

ELSE BEGIN

hl := depth(bt^.lchild);

hr := depth(bt^.rchild);

IF \_\_\_\_\_

THEN \_\_\_\_\_

ELSE depth := hr + 1

END

END;

/\* C 语言, PASCAL 考生不答 \*/

int depth(bitree bt) /\* bt 为根结点指针 \*/

{ int hl, hr;

if (bt == NULL) return (\_\_\_\_\_);

hl = depth(bt->lchild);

hr = depth(bt->rchild);

if (\_\_\_\_\_)

return (hr + 1);

}

12. 已知 N 元整型数组 a 存放 N 个学生的成绩，已按由大到小排序，以下算法是用对分（拆半）查找方法统计成绩大于或等于 x 分的学生人数，请填空使之完善。

{ PASCAL 语言, C 语言考生不答 }

CONST N = 学生人数;

TYPE

```
score = 0 .. 100;  
mat = ARRAY [1..N] OF score;  
FUNCTION uprx (VAR a : mat; x : score) : integer;  
{ 函数返回大于或等于 x 分的学生人数 }  
VAR head, mid, rear : integer;  
BEGIN  
    head := 1; rear := N;  
    REPEAT  
        mid := (head + rear) div 2;  
        IF x <= a [mid]  
            THEN _____  
            ELSE _____  
    UNTIL _____;  
    IF a [head] < x  
        THEN uprx := head - 1  
        ELSE uprx := head  
    END;
```

/\* C 语言, PASCAL 语言考生不答 \*/

#define N 学生人数

```
int uprx (int a [N], int x) /* 函数返回大于或等于 x 分的学生人数 */  
{ int head, mid, rear;  
    head = 0; rear = N - 1;  
    do { mid = (head + rear) / 2;  
        if (x <= a [mid])  
            _____;  
        else _____;  
    } while ( _____ );  
    if (a [head] < x) return head - 1;  
    return head;  
}
```

## 二、简答题 (共 46 分)

1. 有 5 个元素入栈次序为: A、B、C、D、E, 在各种可能的出栈次序中, 以元素 C、D 最先出栈 (即 C 第一个且 D 第二个出栈) 的次序有哪几个。(6 分)

2. 已知 KMP 串匹配算法中，子串为 babababaa，写出 next 数组与改进后的 next 数组信息值（要求写出数组下标起点）。（6 分）
3. 设某通讯电文由 A、B、C、D、E、F 六个字符组成，它们在电文中出现的次数分别是 16, 5, 9, 3, 20, 1。试画出编码用的哈夫曼树。（6 分）
4. 试分析执行下面两段程序后，变量 a 的值（用含变量 n 的数学式表示，其中 n 为正整数）。（6 分）

(1) { PASCAL 语言 }	/* C 语言 */
a:=0;	a=0;
FOR i:=0 TO n DO	for(i=0;i<=n;i++)
FOR j:=0 TO i+1 DO	for(j=0;j<=i+1;j++)
a:=a+1;	a++;
(2) { PASCAL 语言 }	/* C 语言 */
a:=0;i:=0;j:=n;	a=0;i=0;j=n;
WHILE i<=j DO	while(i<=j)
BEGIN	{
a:=a+i+j;	a+=i+j;
i:=i+1;j:=j-1	i++;j--;
END;	}

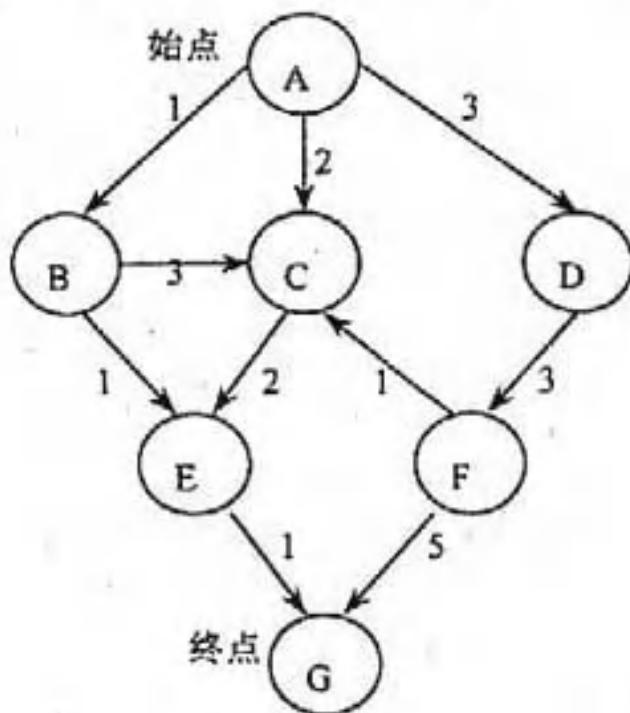
5. 请把如下数据手工构造成一棵平衡二叉排序树并画图。（4 分）  
(23, 76, 47, 53, 41, 12, 85, 30)
6. 已知整型数组 a 的 10 个元素为 326, 129, 167, 588, 212, 95, 980, 725, 443, 501, 用以下排序方法进行由小到大排序。（6 分）

- (1) 用基数排序算法时，试写出第一次分配和收集后数组 A 中的结果。
- (2) 用堆排序算法时，试写出将第一个选出的数据放在数组 a 的最后位置上，将 a 调整为堆之后的 a 中结果。

7. 某赋权有向图如下：

- (1) 试写出深度优先搜索顺序。
- (2) 画出深度优先生成树
- (3) 将该图作为 AOE 网络图，试写出 C 的最早发生时间及活动 FC 的最晚开始时间。

(4) 用 Dijkstra 算法思想计算源点 A 到各顶点的最短路径。



### 三、算法设计（共 34 分）

要求：所有算法以过程或函数形式给出；考生请注明所用语言（PASCAL、C 或 C++）；每个算法应给出实现细节，包括形参定义、变量定义等，并以能够编译通过为标准。

1. 已知数组线性表数据类型定义如下，写一个算法，删除线性表中小于 0 的所有元素。（8 分）

```

CONST maxlen = 线性表最大允许长度 ;    { PASCAL 语言 }
TYPE listp = RECORD
    elem : ARRAY[1..maxlen] OF integer ;
    last : 0..maxlen    { 线性表实际长度 }
END ;
    
```

```

#define maxlen 线性表最大允许长度 /* C 语言 */
typedef struct { int elem[maxlen];
    int last; /* 线性表实际长度 */
} listp ;
    
```

2. 写一个递归算法实现字符串逆序存贮，要求不另设串存贮空间。（8 分）



要求写出主程序第一次调用该递归子程序的格式及参数。

(PASCAL 语言可以使用 TurboPascal 扩充的 string 类型或给出自定义串类型答题)

3. 写一个算法，将一个结点值无序且无重复结点值的单链表 ha 归并到一个结点值按升序连接且无重复结点值的单链表 hb 中，使归并后的单链表各结点值保持升序且没有重复结点值。(要求利用原链表结点存储空间构造归并后的链表，多余的结点则被删除)

已知单链表带附加头结点 (题中 ha, hb 为两链表附加头结点指针)，链表结点结构数据类型定义如下：(9 分)

```
TYPE  pointer = ^ node ;      { PASCAL 语言 }
      node = RECORD  data : integer ; next : pointer  END ;
```

---

```
typedef struct node  /* C 语言 */
```

```
    { int  data ; struct node  *next ;
      } *pointer ;
```

```
#define  new (p)  p = (pointer) malloc ( sizeof ( struct node ))
```

```
#define  dispose (p)  free ((void *)p)
```

```
/* C 语言答题时，请使用以上两个宏定义建立和删除结点 */
```

```
/* C++ 答题时，请使用 new 、 delete 建立和删除结点 */
```

4. 若二叉树采用三叉链表存贮，已知根结点指针为 Bt，写一个不使用堆栈的非递归算法按中序遍历次序打印各结点值。(9 分)

三叉链表结点结构如下：

```
TYPE  thtree = ^ node ;      { PASCAL 语言 }
      node = RECORD  data : char ; parent , lchild , rchild : bitree  END ;
      { parent 为双亲结点指针，根结点的 parent 域为 NIL }
```

---

```
typedef struct node  /* C 语言 */
```

```
    { char data ; struct node  *parent , *lchild , *rchild ;
      } *thtree ;
```

```
/* parent 为双亲结点指针，根结点的 parent 域为 NULL */
```