

考试科目：数据结构

（数据结构部分）（25 分）

一、填空题（1×20 分）

1. 8 层完全二叉树至少有 128 个结点；拥有 100 个结点的完全二叉树其最大层号为 7。

解析：k 层完全二叉树至少有 2^{k-1} 个结点，8 层完全二叉树至少有 128 个结点，完全二叉树最大深度为 $\lceil \log_2 n \rceil + 1$ 。

2. 基于关键字比较大小的排序算法中，快速 排序算法的平均时间复杂度最优。

解析：简单选择排序，直接插入排序和冒泡排序的平均情况下的时间复杂度为 $O(n^2)$ ，快速排序平均性能可达到 $O(\log_2 n)$ ，在实际应用中常常优于其他排序算法。

3. N 个顶点有向图，至少需要 n 条弧才能保证是连通图。

解析：n 个顶点的有向图构成一个环时，有向图为连通的，且边数最少。

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 3。

解析：快速排列之后的序列为：6 15 11 16 32 30
故 16 在数组中的下标位置为 3。

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 'SXSSXSXX'。

解析：1 先栈接着出栈，2，3 入栈，3 出栈，4 入栈再出栈，最后 2 出栈，故 S、X 的操作串为 'SXSSXSXX'。

6. 中缀式 $a+b*3+4*(c-d)$ 对应的前缀式为 $++a*b3*4-cd$ ，若 $a=1$ ， $b=2$ ， $c=3$ ， $d=4$ 则后缀式 $db/cc*a-b*+$ 的运算结果是 18。

解析：表示成前缀式将符号往前移， $a+(b*3)+4*(c-d)=a+(*b3)+[*4(c-d)]=a+(*b3)+(*4-cd)=+(*b3)+(*4-cd)=++a*b3*4-cd$

db/cc*a-b*+ 后缀运算：中缀表达式： $d/b+(c*c-a)*b=18$

7. 用下标从 0 开始的 N 元素组实现循环队列时，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)；

$m=(m+1)\%N$ (填 C 语言，PASCAL 考生不填)。

解析：当 m+1 后超出了数组的有效下标，当 mod N 时，下标又回到 0，不会超出数组的有效下标。

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 $O(n)$ 。

解析：n 个顶点的非平衡二叉树排序树最坏情况下为一棵 n 个结点的单分支树，故最坏查找的平均时间复杂度为 $O(n)$ 。

9. 以下程序的功能是实现带附加头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(____p!=MULL____)
    {q=p; p=p->next;
        q->next=h->next; h->next=____q____;
    }
}
```

解析：该题是带头结点的单链表的逆置，采用头插法将链表逆置。

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; ichild, 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 yst(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*/

p=p->lchild;

}

else

{p=Pop(s); /*栈顶元素出栈*/

printf("%c ",p->data);

p=p->rchild

}

}

解析：二叉树中序遍历非递归算法。

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，请填空并完善之。

{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 语言考生不答*/

int depth(bitree bt)/*bt 为根结点指针*/

{int hl,hr;

if(bt==NULL) return(0);

hl=depth(bt->lchild);

hr=depth(bt->rchild);

if(hl>hr;) hr=hl;

return(hr+1);

}

解析：递归求二叉树的深度算法。

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（拆分）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)div2;

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 分学生人数*/

{int head,min,rear;

head=0;rear=N-1;

do{mid=(head+rear)/2;

if(x<=a[mid])

head=mid+1;

else rear=mid-1;

}while(head<rear);

if(a[head]<x)return head-1;

return head;

}

二、简答题（共 48 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）

解析：CDEBA; CDBEA; CDBAE

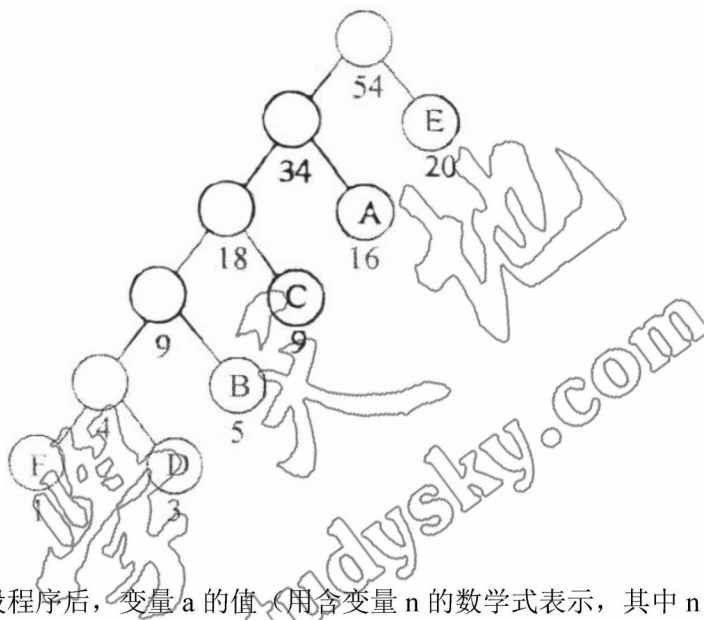
2. 已知 KMP 串匹配算法中，子串 babababaa，写出 next 数组与改进后的 next 数组信

息值（要求写出数组下标起点）。（6分）

解析：next[j]=0 1 2 3 4 5 6 7，下标起点为 j=1，改进后 next[j]=0 1 0 1 1 1 2 3 4 7，下标起点为 j=1。（见教材 P81-84）

3. 设某通讯电文由 A、B、C、D、E、F 六个字符组成，他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）

解析：哈夫曼树如下：



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}

```
a:=0;
FOR i:=0 TO n DO
  FOR j:=0 TO i+1 DO
    a:=a+1;
```

(2) {PASCAL 语言}

```
a:=0;i:=0;j:=n;
WHILE i<=j DO
  BEGIN
    a:=a+i+j;
    i:=i+1;j:=j-1
  END
```

{C 语言*}

```
a=0
for(i=0;i<=n;i++)
  for(j=0;j<=i+1;j++)
    a++;
```

{C 语言*}

```
a:=0;i:=0;j:=n
while(i<=j)
{
  a:=a+i+j;
  i++;j--;
}
```

解析：(1) $\frac{(n+1)(n+4)}{2}$

解析： $\sum_{i=0}^n \sum_{j=0}^{i+1} 1 = \sum_{i=0}^n (i+2) = \frac{(n+1)(n+4)}{2}$

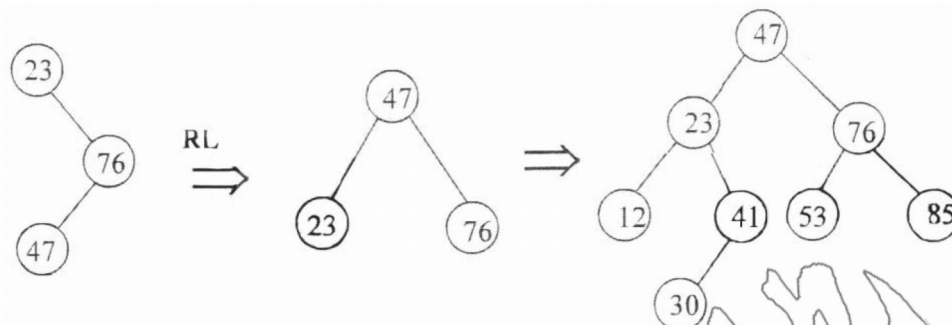
(2) $n \left\lceil \frac{n+1}{2} \right\rceil$

解析：while 循环共执行 $\left\lceil \frac{n+1}{2} \right\rceil$ ，i+j 的值始终为 n。

5. 请把如下数据手工构造一颗平衡二叉排序树并画图。（4 分）

(23, 76, 47, 53, 41, 12, 85, 30)

解析：



6. 已知整型数组 a 的 10 个元素为 326, 129, 167, 588, 212, 95, 980, 725, 443, 601，用一下排序法进行由小到大的排序。（6 分）

(1) 用基数排序算法时，试写出第一次分配后数组 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。

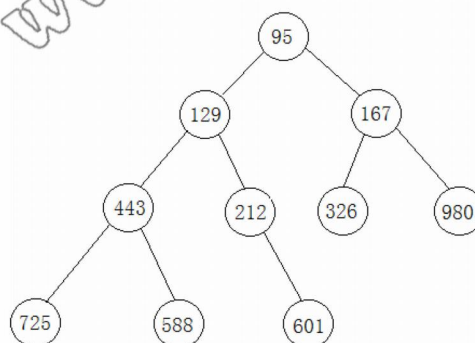
解析：(1 基数排序)

第一次分配

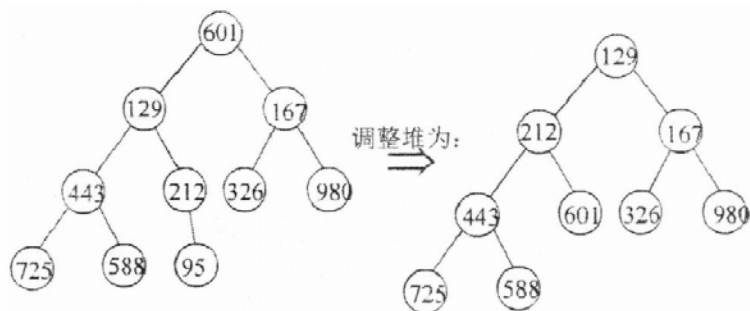
0	1	2	3	4	5	6	7	8	9
					725				
908	601	212	443		95	326	167	588	129

第一次收集：980, 601, 212, 443, 95, 725, 326, 167, 588, 129

(2) 首先调整堆：



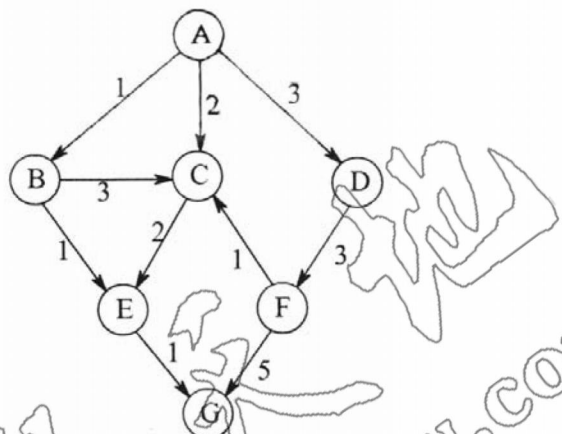
输出 95 放到 a 的最后位置，即与 601 交换



a 中结果：129,212,167,443,601,326,980,725,588,95

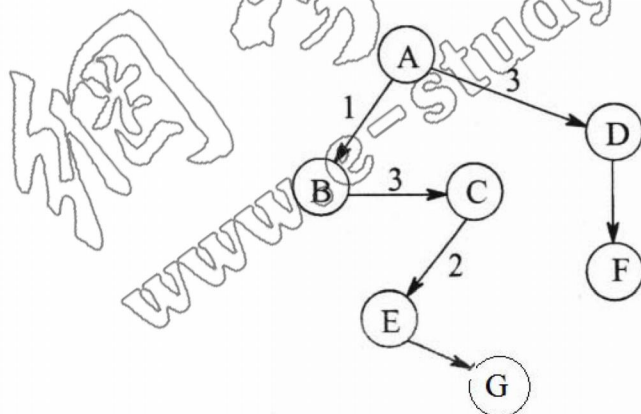
7、某赋权有向图如下：

- (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 的最晚时间。
- (4) 用 Dijkstra 算法思想计算源点 A 到各处顶点的最短路径。



解析：(1) 深度优先搜索顺序：ABCEGDF

(2) 深度优先生成树：



(3) C 的最早发生时间：7 活动 FC 的最晚开始时间：7

解析：A 的最早发生时间：0 B 的最早发生时间：1 F 的最早发生时间：6

∴ C 的最早发生时间 $\max\{2,4,7\}=7$

	A	B	C	D	E	F	G
时间的最早发生时间：	0	1	7	3	6	6	11
时间的最晚发生时间：	0	5	8	3	10	6	11

∴ FC 的最晚开始时间：VI (c) -1=8-1=7

(4) Dijkstra 算法计算最短路径

顶点	第一趟	第二趟	第三趟	第四趟	第五趟	第六趟
B	{A,B}1					
C	{A,C}2	{A,C}2	{A,C}2			
D	{A,D}3	{A,D}3	{A,D}3	{A,D}3		
E	∞	{A,B,E}2				
F	∞	∞	∞	∞	{A,D,F}6	{A,D,F}6
G	∞	∞	{A,B,E,G}3	{A,B,E,G}3	{A,B,E,G}3	
集合 S	{A,B}	{A,B,B}	{A,B,E,C}	{A,B,E,C,D}	{A,B,E,C,D,G}	{A,B,E,C,D,G,F}

故 A 点到各顶点的最短路径为：

A→B: 1 A→C: 2 A→D: 3 A→E: 2 A→F: 6 A→G: 3

三、算法设计（共 34 分）

要求：所有算法以过程或函数形式给出：考生请注明所有语言（PASCAL、C 或者 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 listtp=RECORD

elem:ARRAY[..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*/

}listtp;

解析：

1、

#define maxlen x 线性表最大允许长度

typedef struct{int elem[maxlen];

int last /* 线性表实际长度*/

}listtp;

listtp Delete(listtp list)

{

for(int i=0;i<list.last;

{

if(list.elem[i]<0)

{


```
list.elem[i]=list.elem[i+1];
continue;
}
i++;
}
list.last=i+1;
return list;
}
```

2. 写出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）

解析：

```
#include "stdio.h"
#include "string.h"
//算法如下
void strorder(char*str,int len)
{
    char temp;
    for(int i=0;i<len/2;i++)
    {
        temp=str[i];
        str[i]=str[len-1-i];
        str[len-1-i]=temp;
    }
}
int main (int argc,char*argv[])
{
    //测试算法，以及应用上述算法函数的格式以及参数
    char str[]="hello*";
    int len=strlen(str);
    strorder(str,len);
    printf( "%s\n",str);
    return 0;
}
```

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贮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 为量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  pointe=^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 */
```

解析：

```
Linklist Merge(Linklist ha,linklist hb)
{
    LNode*p=ha->next,*q=hb->next,r=hb;
    while(p && q)
    {
        if(p->data==q->data)
        {u=p,p=p->next;free(u);}
        else if(p->data<q->data)
        {s=p->next;p->next=q;r->next=p;r=q;
        q=q->next;p=s;}
        else
        {r=q;q=q->next;}
        q=hb->next;
    }
    Free(ha);
    return hb;
}
```

4.若二叉树采用三叉链表存贮，已知根据点指针为 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 three=^node {PASCAL 语言}
node=RECORD data:char;parent,lchild:bitree END;
{parent 为双亲结点指针，根据点的 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，根据点的 parent 域为 NULL*/
```

解析：

```
void inorder(Bit7Node*bt)
{
    Bit7Node*ps;
    if(bt==Null)
    return;
    p=bt
```

```
while(p)
{if(p→lchild)p=p→lchild;
else
visit(p→data);
if(p→rchild)p=p→rchild;
else
{while(1)
{ps=p
p=p→parent;
if(p==Null)
break;
if(p→lchild==ps & & p→rchild)
break;}
}
if(p)p=p→rchild;}
}
```
