

西南交通大学 2001 年硕士研究生招生入学试题解析

考试科目：数据结构

考生请注意：

4. 本试题共五题，共 4 页，考生请认真检查；
5. 答题时，直接将答题内容写在试题和由我校提供的答题纸上；
6. 本试题不得拆开，拆开后遗失后果自负。

题号	一	二	三	四	五	六	七	八	九		总分
得分											
签字											

说明：考生可以选用类—OASCAL 或类—C 语言答题。判断题、选择题和填空题可以直接答在试题上，简答题和算法设计题答在答题纸上。

一、判断题，正确的答√，错误的打×（共 10 分，每题 2 分）

1. KMP 算法的最大特点是知识主串的指针不需要回溯。（√）

解析：见课本 P80，KMP 算法解析。

2、一颗有 n 个结点的二叉树，从上到下，从左到右用自然数依次给与编号，则编号为 i 的结点在左孩子的编号是 $2i$ ($2i \leq n$)，右孩子是 $2i+1$ ($2i+1 \leq n$)（×）

解析：应该是对于完美二叉树而言的，不是全部的二叉树。

3、有 e 条边的无向图，在邻接表中有 e 个结点（×）

解析：应该有 $2e$ 个结点。

4、任何有向网络（AOV-网）拓扑排序的结果是唯一的。（×）

解析：排序结果不一定唯一。

5、在栈空的情况下，不能做退栈运算，否则产生下溢。（√）

解析：在出栈的时候要先判栈空，否则会产生下溢现象。

二、选择题（共 20 分，每题 2 分，A、B、C、D 中只有一个是正确或最确切的）

1、在一个以 h 为头的单循环链中， p 指针指向链尾的条件是（A）

A) $p \rightarrow \text{next} = h$ B) $p \rightarrow \text{next} = \text{NIL}$ C) $p \rightarrow \text{next} \rightarrow \text{next} = h$ D) $p \rightarrow \text{data} = -1$

解析：单循环链表的尾结点指向头结点，若 P 为尾结点，则 $P \rightarrow \text{next} = h$

2、设字符串 $S = \text{'ABCDEF'}$ ， $T = \text{'PQRST'}$ ，则运算

CONCAT(SUB(S,2,LENGTH(T),SUB(S,LENGTH(T),2)))后的结果为（D）

A) 'BCQR' B) 'BCDEF' C) 'BCDEFG' D) 'BCDEFEF'

解析：LENGTH(T)用来求串的长度，SUB($s,2,\text{LENGTH}(T)$)指从 s 串中的第二个字符开始截取 T 的长度个字符，即“BCDEF”，SUB($s,2,\text{LENGTH}(T)2$)指从 s 串中第五个字符开始截取两个字符（这里 T 的长度为 5），即“EF”，CONTACT 指联接两个串，即“BCDEFEF”。

3、下面序列中，（B）是堆

A)1, 5, 10, 6, 7, 8, 9, 2

B)1, 2, 8, 4, 3, 9, 10, 5

C)9, 8, 7, 4, 6, 8, 2, 1

D)9, 8, 7, 6, 5, 4, 3, 7

解析：按层次序列排列各组数据，B 为小根堆，根结点小于左右孩子结点的值

4、以二叉链表为存储结构的二叉树 D，没有左孩子的主要条件是：（ C ）

A) D->lchild=NULL

B) D->rchild=NULL

C) D->ltag=1

D) D->ltag=0

解析：线索二叉树中，标志域含义，Itag $\begin{cases} 0lchild \text{域指示结点的左孩子} \\ 1lchild \text{域指示结点的前驱} \end{cases}$

5、设有数组 A[i,j]，数组的每个元素长度为 3 字节，i 的值为 1 到 8，j 的值为 1 到 10，数组从内存地址 BA 开始顺序存放，当以行主存放时，元素 A[5,8] 的存储首地址为（ A ）。

A) BA+141

B) BA+180

C) BA+222

D) BA+225

解析：以行序列为主序， $\text{locA}[5][8] = \text{BA} + [(5-1) \times 10 + (8-1)] \times 3 = \text{BA} + 141$

6、设 abcdef 以所给的次序进栈，若在进栈操作是，允许退栈操作，则下面得不到的序列为（ D ）

A) fedcba

B) bcafed

C) dcefb a

D) cabdef

解析：栈是一种操作受限的线性表，“先进后出”，D 中 a 不可能比 b 先出栈

7、若数组 S[1..n] 作为两个栈 S1 和 S2 的存储空间，对任何一个栈，只有 [1..n] 当全满时才能进行进栈操作，为这两个栈分配空间的方案是（ C ）

A) S1 的栈底位置为 0，S2 的栈底位置为 n+1

B) S1 的栈底位置为 0，S2 的栈底位置为 n/1

C) S1 的栈底位置为 1，S2 的栈底位置为 n

D) S1 的栈底位置为 1，S2 的栈底位置为 1

解析：为了有效的利用存储空间，采用共享栈的思想，将两个栈的栈底设在两端

8、若线性表最常用的操作是存取第 i 个元素及其前趋的值，则采用（ D ）存储方式节省时间

A) 单链表

B) 双链表

C) 单循环链表

D) 顺序表

解析：要取第 i 个元素的及其前驱的值，若找其前驱，指针需移到链表末尾，用双链表直接就可以招待其前驱的值。

9、在平衡二叉树中插入一个节点后造成了不平衡，设最低的不平衡结点为 A，并已知 A 的左孩子的平衡因子 -1，右孩子的平衡因子为 0，则应作（ A ）型调整以使其平衡。

A) LL

B) LR

C) RL

D) RR

解析：见课本 P234

10、若给定的关键字集合为 {20,15,14,18,21,36,40,10}，一趟快速排序结束时，数据的排列为（ A ）

A) 10,15,14,18,20,36,40,21

B) 10,15,14,18,20,40,36,21

C) 10,15,14,20,18,40,36,21

D) 15,10,14,18,20,36,40,21

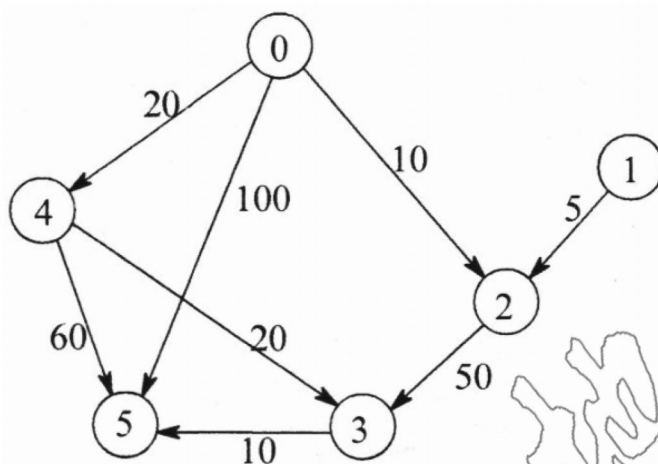
解析：这是快速排序算法，将第一个关键字作为枢轴值，比枢轴值的元素向右移动，比枢轴值小的往左移动，表中的元素倍数轴值一分为二。

三、简答题（共 20 分，每题 4 分）

1、有向图 $G = \langle V, E \rangle$ ，其中 $V = \{0, 1, 2, 3, 4, 5\}$ ，用 $\langle a, b, d \rangle$ 三元组表示弧 $\langle a, b \rangle$ 及弧上的权为

$E=\{<0,5,100>,<0,2,10>,<1,2,5>,<0,4,20>,<4,5,60>,<3,5,10>,<2,3,50>,<4,3,20>\}$ ，则从源点 0 到顶点 3 的最短路径长度是多少？经过哪些中间顶点？

解析：根据三元组中的顶点及权作图



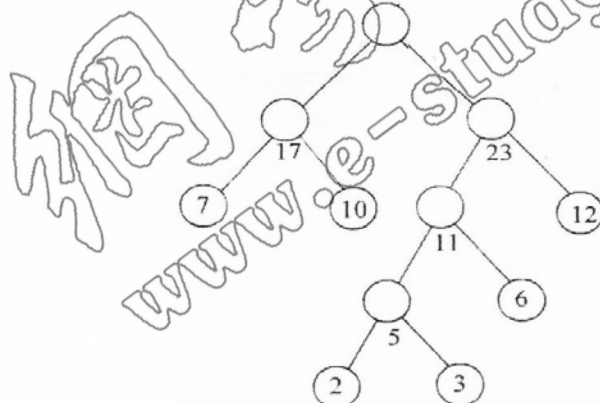
0→3 的路径长度：{0,4,3}20+20=40

{0,2,3}60

故最短路径长度是 40，经过的中间结点为 4

2、给定一组数据（6,2,7,10,3,12），以他构造一颗哈夫曼树，请求出树高及树的带权路径长度 wpl 的值。

解析：哈夫曼的构造过程件课本 P145,哈夫曼树如下：



树高为 5， $wpl = (7+10+12) * 2 + 6 * 3 + (2+3) * 4 = 96$

3、设有 4000 个无序的元素，希望用最快速度挑选出前 10 个最大的元素，在一下的排序方中，采用哪种方法最好？为什么？（快速排序、堆排序、基数排序）

解析：用堆排序最好

快速排序和基数排序都需要将所有无序元素排序之后才能跳线出前十个最大元素，而堆排序建好堆之后每一趟排序即可挑选出一个最大元素。

4、假定有 n 个关键字，具有相同的散列函数值，如果用线性探测法把这 n 个关键字放到散列表中，需做多少次探测？

解析：需要做探测的次数： $1+2+3+\dots+n=n*(n+1)/2$ 次

5、设循环列 Q[0...N-1]的头尾指针 F，R，当插入元时素时尾指针 R 加 1，头指针 F 总是指队列中第一个

元素的前一个位置，试求该队列中元素的个数。

解析：队列中元素的个数为： $(R-F+N) \% N$ 用队尾指针与队头指针的差来表示元素个数

四. 算法填空题（共 22 分，每空 2 分，下面算法分别用类—PASCAL 或类—C 语言给出，考生可以任选一个做，都不选不给分）

1、以下算法是完成将单向循环链表 F 就地改为双向循环链表的功能，请填空使之完善。说明：链表 F 的结点结构为

pre	data	next
-----	------	------

其中 data 为数据域，next 和 pre 为指针域，且 next 域的值值为后序结点的地址，

pre 空

[类—PASCAL]

PROC Double_List(F);

{p,q 均为移动指针}

IF F' , next=F

THEN[F' pre:=_____;

RETURN];

q :=F p:=F' .next;

REPEAT

p' .pre:=_____ q:=_____;

_____:=p' .next

UNTIL p=_____;

p' .pre:=q

ENDP:{Double_List}

{类—C}

void Double_List(F)

//p,q 均为移动指针

{if (F->next==F)

{F->pre=_____;

return};

q=F;p=F->next;

do{

p->pre=_____q=_____;

_____:=p->next;

}while(p=_____);

p->pre=q;

//Double_List

解析：1、【类-c】

F;q; q→next; P; F

2.下面算法是完成在二叉排序树 T 中查找关键值为 k 的结点，请填空使之完善。二叉排序树的结点类型入下：

[类—PASCAL]

TYPE bireptr=^node;

node=RECORD

data:char;

lchild,rchild:bireptr

FUNC Search(T,k): bireptr{成功时返回指向该结点的指针，否则返回空指针}

IF T=NIL THEN RETURN(NIL)

ELSE IF T^data=k THEN RETURN(T)

ELSE IF T^data>k THEN _____

ELSE _____;

ENDF:{Search}

{类—C}

typedef struct Node{

char data;

struct Node *lchild,*rchild;

}Node,*Bitree;

Bitree*Search(T,k)//成功时返回指向该结点的指针，否则返回空指针

{if (! T) return Null;

else if (T->data==k)return T;

else if (T->data>k)_____;

else _____;

//Search

解析：【类-c】

Return Search (T→lchild, k);

Return Search (T→rchild, k);

3、下面算法是以有向图的邻接表为基础，查找顶点 k 的入度算法，请填空使之完善。

说明：adj 为邻接表，n 为表的顶点数，结点的两个域为顶点域 vex,指针域 nextarc.邻接表指针域为 firstarc

类—PASCAL

```
FUNC find (k,adj) :integer;
Count:=0
FOR i:=1 TO n DO
  [p:=_____;
  WHILE ____ DO
    [IF _____
      THRN Count:=Count+1;
      P:=_____]
  [
  RETURN(Count)
  ENDF:{find}
```

类—C

```
Void find(k,adj)
{Count=0;
for(i=1;i<=n;++i)
{p=_____;
while _____
{if _____
  Count++;
  p=_____;}
}
return Count;
} //find
```

解析：【类-c】

Vex[i].firstarc;P!=NULL;P→data==vex[k];P→nextarc;

五、算法设计题（共 28 分。要求：所有算法以过程或函数形式给出，并同时给出设计思想及必要的中文注释：考生请注明所有语言（类—PASCAL 或类—C）。

1、已知 N 元整型数组 a 存放 N 个学生成绩，已按由大到小排序，试设计一个算法，用折半查找方法统计成绩大于或等于 x 分的学生人数。（8 分）

解析：

#define N

Int uprx(int a[N],int x)

{int head=1,mid,rear=N;

do{

mid=(head+rear)/2;

if(x<=a[mid])

rear=mid-1

else head=mid+1

}while(head>rear);

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

Return head;

{

2.设一颗二叉树 T 以二叉链表作为存储结构，试编写将二叉树 T 中所有节点的左、右子树相互交换的算法。（10 分）

解析：

void swap(Bitree T)

{if(T!=NULL)

{swap(T→lchild);

swap(T→rchild);

temp(T→lchild);

T→lchild= T→rchild;

T→rchild=temp;

{

{

3. 已知一个线性表中元素均为正、负整数，且一次存储在数组 $A[0...n-1]$ 中，试设计算法将表中所有正整数均排列在负整数之后。要求不另外增加存储空间，且时间复杂度为 $O(n)$ 。

解析：

```
void move(sqlist& L)
```

```
{Elem Type temp;
```

```
Int i=0,j=N-1
```

```
while(i<j)
```

```
{while(i<j && L.A[i]<0)
```

```
i++;
```

```
while(i<j && L.A[j]>0
```

```
j--;
```

```
if(j<j)
```

```
{temp=L.A[i];
```

```
L.A[i]=L.A[j];
```

```
L.A[j]=temp;
```

```
}
```

```
}
```

```
}
```