

西南交通大学 2004 年硕士生入学考试试题

试题名称: 数据结构

考生注意:

1. 本试题共 四 题, 共 六 页, 请考生认真检查;
2. 请务必将答案写在答卷纸上, 写在试卷上的答案无效。

题号	一	二	三	四	总分
得分					
签字					

一、 填空与选择题 (20 分)

1) 数据结构与数据类型的形式定义分别为:

Data-Structure=(D, R), (b, d)

Data-Type=(D, R, P), (c, d, f)

试选择 D, R, P 的确切含义。

- a) 数据 b) 数据元素 c) 数据对象
 d) 关系 e) 存储结构 f) 基本操作

2) 有 n 个节点的二叉树的最大深度为 n , 最小深度为 $\lfloor \log_2 n \rfloor + 1$;
 有 n 个节点的完全二叉树的最大深度为 $\lfloor \log_2 n \rfloor + 1$ 。

3) 图书馆要对成千上万册的书籍进行计算机管理, 每本书包含若干信息。从数据结构的观点出发, 请在下列的关键词中选择一合适的词作为数据元素的单位:
 a) 出版社名 b) 书名 c) 作者名 d) 目录 e) 一本书 f) 一页。

4) 排序二叉树查找的最坏时间复杂度是 $O(n^2)$; 平衡二叉树查找的最坏时间复杂度是 $O(\log n)$ 。

5) 要从 10000 个数据元素中选 10 个最小的, 你将选择下面哪种排序方法设计该算法?

- a) 希尔排序方法; b) 快速排序方法;
 c) 堆排序方法; d) 简单选择排序方法;
 e) 冒泡方法。

7) 已知 L 是有表头结点的非空循环单链表，试从下列提供的答案中选择合适

- a) 删除 P 结点之后的结点的语句序列是 (2)(5)(9) ;
 b) 在 P 结点前插入 S 结点的语句序列是 (6)(8)(4)(1) .

- (1) P->next=S ;
 (2) Q=P->next ;
 (3) P->next=S->next ;
 (4) S->next=P->next ;
 (5) P->next=Q->next ;
 (6) Q=P ;
 (7) P=Q ;
 (8) while (P->next != Q) P=P->next ;
 (9) free (Q) .

8) 算法设计中，对算法有哪四个基本要求？ 正确性，可读性，健壮性，时间与低存储需求。

9) 基本操作是数据类型的重要组成部分，试列出六种以上的基本操作名称。
; ; ; ; ; ; ;

10) 请在下列的答案中选择可能正确的答案。

栈的操作特点是： bc； (不需全选)

队列的操作特点是： ad

- a) 先进先出 b) 后进先出
 c) 先进后出 d) 后进后出

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
模式串	A	B	A	A	A	C	D	A	B	A	A	A	C	D	A
next	0	1	2	2	2	1	1	2	3	4	2	2	1	1	

二 基础题 (50 分)

1) 设模式串 pat='ABAAACDABAAACDA'，求 pat 的 next[j]。

2) 设有解决同一问题的两个算法 A 与 B，时间复杂度分别为：

$f_A(n)=(n-20)^2$; $f_B(n)=n+90$ 。

试分析规模 n 在什么范围内，算法 A 优于算法 B，为什么？

$10 < n < 31$ 时， $f_A(n) < f_B(n)$ 。

3) 设有如下 $2n \times 2n$ 的特殊矩阵 A，将其压缩存储到一维数组 SA 中

A =

a_{11}	a_{12}		
a_{21}	a_{22}		
		a_{33}	a_{34}
		a_{43}	a_{44}

$k = \begin{cases} 2i-1, & i=j+1 \\ 2i, & i=j \\ 2i, & i=2j \end{cases}$

$$a_{11} a_{12} a_{21} a_{22} a_{33} \dots a_{2n-1} a_{2n} a_{2n-1} a_{2n} a_{2n} a_{2n-1} a_{2n} a_{2n}$$

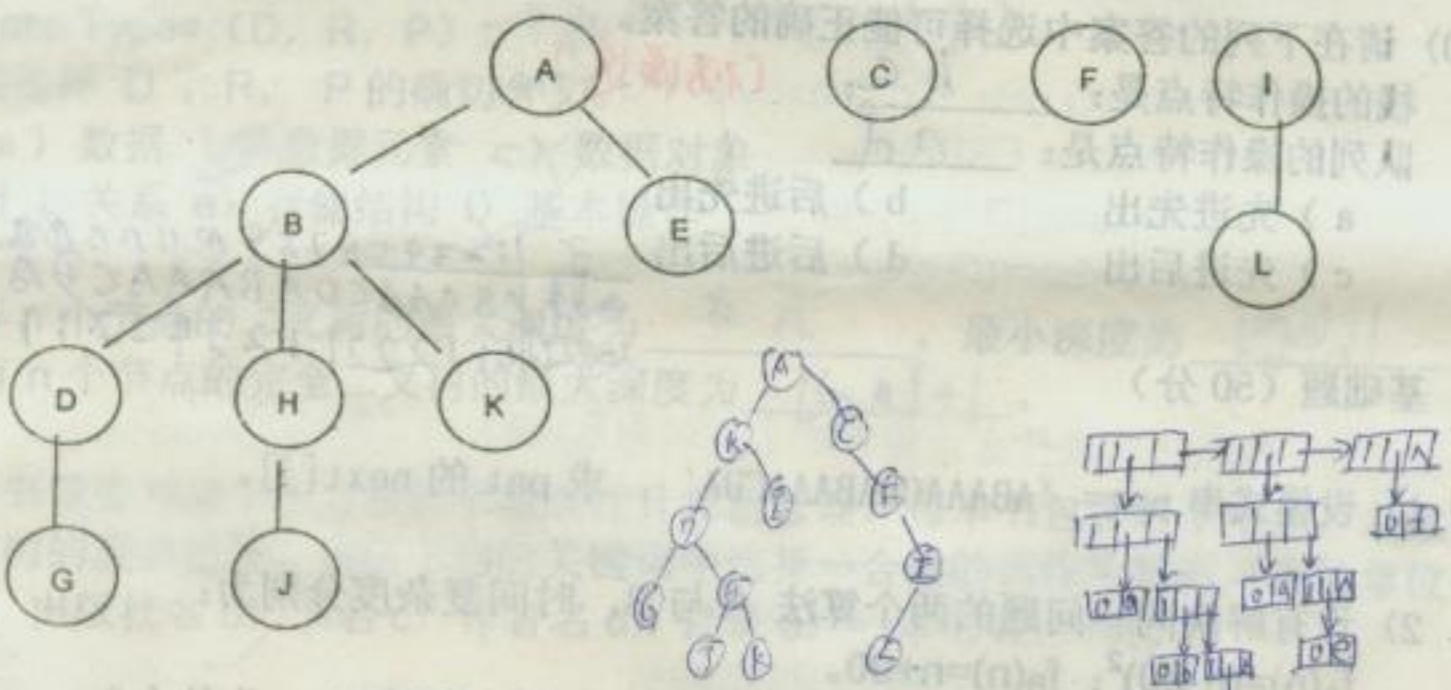
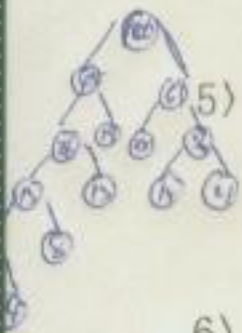
若 $A[i, j]$ 为非零元素, 写出由下标 $[i, j]$ 求 k 的转换公式。

4) 已知一棵度为 3 的树中, 有 8 个度为 2 的结点, 5 个度为 3 的结点, 问该树中有多少个叶子结点? $n_0 = 2n_2 + 3n_3 - 1$

5) 假设用于通信的电文仅由 8 个字母构成, 字母在电文中出现的频率分别为 0.15, 0.19, 0.04, 0.08, 0.25, 0.01, 0.21, 0.13。试为这 8 个字母设计哈夫曼编码。

6) 假设一棵二叉数的先序序列为 $abdficegh$, 中序序列为 $bfidagehc$, 请画出该二叉树。

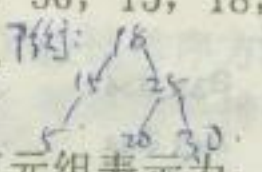
7) 下图为森林, 画出兄弟孩子表示法表示的二叉树。



8) 设有广义表 $A = ((a, (b, c)), (d, e), f)$

- a) 该表的长度是多少? 3
- b) 画出该表的链式存储结构。

9) 按如下序列的给定顺序: 20, 25, 30, 15, 18, 5 画出平衡二叉树以及排序二叉树。



10) 假设某 6×6 的稀疏矩阵的三元组表示为:
 $(1, 3, 1), (2, 1, 2), (2, 4, 3), (3, 6, 4),$
 $(4, 2, 5), (4, 5, 6), (5, 2, 7), (5, 3, 8)$

1) 算法分析

```

for (i=1; i<=n; i++)
for (j=1; j<=n; j++)
{
k=1;
while (k<=n) k=5*k
}

```

时间复杂度为 $O(n^2)$

b)

```

BTree SearchBST(BTree T, keytype key)
{
if ((!T) || EQ(key, T->data.key)) return (T);
else if LT (key, T->data.key)
return(SearchBST(T->lchild, key));
else return(SearchBST(T->rchild, key));
}

```

时间复杂度为 $O(\lg n)$

2 > 简述算法功能

```

a) {
if ((Q.rear+1)%Maxsize==Q.front)
return ERROR;
Q.base[Q.rear]=e;
Q.rear=(Q.rear+1)%Maxsize;
return OK;
}

```

该算法实现什么功能? *若队列不满, 将元素e插入队尾.*

```

b) {
if (s.top-s.base >= s.stacksize)
{ s.base=追加分配空间;
s.top=s.base+s.stacksize;
s.stacksize+=Stackincrement;
}
*s.top++=e;
return Ok
}

```

该算法实现什么功能? *入栈操作.*

```

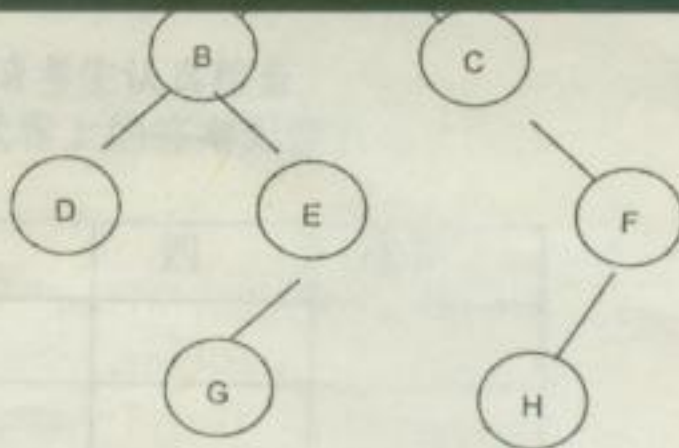
c) {
for (i=0; i<S.length && i<T.length; ++i)
if (S.ch[i]!=T.ch[i])

```

Void visit (T) :

```
{
    visit (T->rchild);
    printf(T->data);
    visit (T->lchild);
}
```

打印顺序



运行结果: DGEBACHF

b) 设 dlt[1]=[5], dlt[0]=[3], L=[5,0,4,6,3,7,2,8,1,9];

运行 Shellsort(L,*dlt, 2) 后,

L=[0 4 2 3 6 5 8 7 9]?

```
void Shellinsert(SqList &L,int dk)
{ for (i=dk+1 ; i<=L.length ; ++i)
  if LT(L.r[i].key, L.r[i-dk].key)
  { L.r[0] = L.r[i];
    for( j=i-dk ; j>0&&LT(L.r[0].key,L.r[j].key) ; j-=dk)
      L.r[j+dk]=L.r[j];
    L.r[j+dk]=L.r[0];
  }
}

void Shellsort( SqList &L,int *dlt, int t)
{ for (k=t; k>0; --k)
  Shellinsert(L, dlt[k-1]);
}
```

4) 算法改错

a) 下面为快速排序中的一次划分算法，找出错误并修改完善该算法；在修改后的算法中，标记出原出错处。

```
int partition(SqList &L,int low,int high)
{ temp= L.r[high];
  while (low<key) high
  { while (low<high &&L.r[high].key>=temp.key) --high;
    L.r[low]=temp; temp=L.r[high]; L.r[low]=L.r[high];
    while (low<high &&L.r[low].key<=temp.key) ++ high;
    L.r[high]=L.r[low];
  }
  L.r[low]=L.r[high]; return low;
```

```

while(pa && L)
{
    if(pa->data == (La->data))
    {
        pc->next = pa; pc = pa; pa = pa->next;
    }
    pc->next = pa?pa:pb;
    free(La);
} //M ergelist
    
```

四、 算法设计 (40 分)

(*****以下部分可以用 c、pascal、类c、类pascal描述算法*****)

- 1) 设计由 n 个学生构成的线性表。假设每个学生包含的信息为：学号，姓名，年龄，籍贯，电话号码，总成绩等。
 - (a) 用高级语言描述数据元素及线性表的存储结构；
 - (b) 给出线性表的抽象数据类型 (ADT) 描述；
 - (c) 设计在线性表中插入一学生纪录的算法。
 - (d) 设计在线性表中删除一学生纪录的算法。

2) 有一链式二叉树 btree，结点结构为 (lchild, data, rchild)。分别设计如下算法 (可以用递归算法)：

- (a) 用高级语言描述链式二叉树的存储结构；
- (b) 计算二叉树中度为 0 的结点数目；(求叶子结点数)
- (c) 计算二叉树的深度；
- (d) 后序遍历二叉树。

```

(a) typedef struct BiTNode {
    TElemType data;
    struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
    
```

```

(b) int leafcount-bit(BiTree T)
{
    if(!T) return 0;
    m = leafcount-bit(T->lchild);
    n = leafcount-bit(T->rchild);
    if(!m && !n) return 1;
    else return (m+n);
}
    
```

```

(c) int Get-depth(BiTree T)
{
    if(!T) return 0;
    m = Get-depth(T->lchild);
    n = Get-depth(T->rchild);
    return (max(m,n)+1);
}
    
```

```

(d) postorder(BiTree T) {
    if(T) {
        postorder(T->lchild);
        postorder(T->rchild);
        visit(T->data);
    }
}
    
```