

西南交通大学 2002 年全日制硕士研究生入学试题

试题名称：数据结构

一、选择与填空题（12 分）

1. b 用于描述数据对象及数据元素间的关系；而 c 描述了数据对象、数据元素间的关系及数据的基本处理方法。

a) 数据 b) 数据结构 c) 数据类型 d) 存储结构

解析：数据结构用于描述数据元素以及数据元素之间的关系；数据类型描述了数据对象间及关系以及基本操作

2. 二叉树第 K 层至多有 b 个结点。

a) $2K-1$ b) 2^{k-1} c) $2^k - 1$

解析：二叉树的性质

3. 下面算法的时间复杂度为： $O(n^2 \log_5 n)$ 。

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

解析：基本运算语句为 $k=5*k$ ，设为 $T(n)$ ，对 j 每循环一次执行次数为 m ，所以 $5^m \leq n$ ，

即 $m \leq \log_5 n$ ，所以 $T(n) = \sum_{i=1}^n \sum_{j=1}^n m = m \cdot n^2 = n^2 \log_5 n$ ，所以时间复杂度为 $O(n^2 \log_5 n)$

4. 已知 P 结点是某双向链表的中间结点，试从下列提供的答案中选择合适的语句序列。

a) 在 P 结点后插入 S 结点的语句序列是 (7) (12) (1) (6)；

b) 在 P 结点前插入 S 结点的语句序列是 (13) (8) (5) (12)；

c) 删除 P 结点的直接后继结点的语句序列是 (15) (3) (4) (18)；

(1) P->next=S

(2) P->prior=S

(3) P->next=Q->next

(4) Q->next->prior=Q->prior

(5) S->next=P

(6) S->prior=P

(7) S->next=P->next

(8) S->prior=P->prior

(9) P->prior->next=p->next

(10) P->prior->next=P

- (11) $P \rightarrow next \rightarrow priou = P$
- (12) $P \rightarrow next \rightarrow priou = S$
- (13) $P \rightarrow priou \rightarrow next = S$
- (14) $P \rightarrow next \rightarrow priou = P \rightarrow priou$
- (15) $Q = P \rightarrow next$
- (16) $Q = P \rightarrow priou$
- (17) $free(P)$
- (18) $free(Q)$

5. 要从 1000 个数据元素中选出五个最小的，下面排序算法中，哪个算法最快？c。

a) 希尔排序 b) 快速排序 c) 堆排序 d) 简单选择排序

解析：见 01 年三、3 解析

6. $T(n) = O(f(n))$ 中，函数 $O()$ 的正确含义为 c。

a) $T(n)$ 为 $f(n)$ 的函数

b) $T(n)$ 为 n 的函数

c) 存在足够大的正整数 M ，使得 $T(n) \leq M \times f(n)$

7. 非平衡排序二叉树查找的最坏时间复杂度是 $O(n)$ 。

解析：非平衡二叉排序树最坏的情况下为一个单分支，树高等于其结点数，查找时间复杂度为 $O(n)$ 。

二、求解下列各问题（30 分）

1. 已知有实现统一功能的两种算法，时间复杂度分别为 $O(10^n)$ 和 $O(n^{10})$ ，假设计算机可连续运算的时间为 10^{12} 秒，而每秒计算机可执行的基本操作为 10^2 ，试问在此条件下，这两个算法可解决问题的规模（即 n 值得最大值）各为多少？哪个算法更适宜？

解析： $O(10^n) = 14$ ， $O(n^{10}) = 20$

$O(n^{10})$ 更适宜

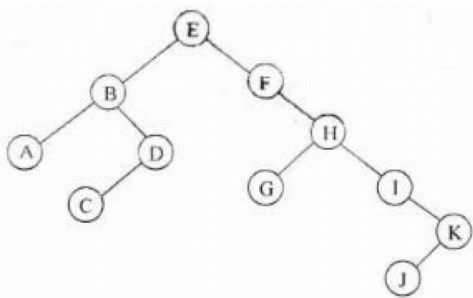
2. 已知在一棵含有 n 个结点的树中，只有度为 K 的分支结点和度为 0 的叶子结点，求该树叶子结点的数目。

解析：叶子结点是数目为 $n_0 = (kn - n + 1)/k$

$$\begin{cases} n = n_2 + n_k (1) \\ n = kn_k + 1 (2) \end{cases}, \text{由 (1) 得 } n_k = n - n_0, \text{代入 (2) 中可得 } n_0 = (kn - n + 1)/k$$

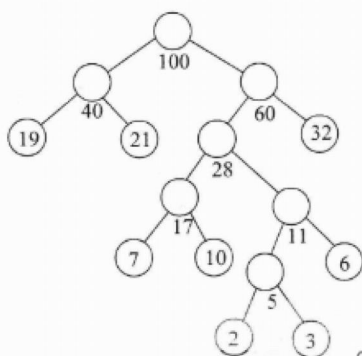
3. 假设一棵二叉树的先序序列为 EBADCFHGIKJ，中序序列为 ABCDEFGHJK，请画出该二叉树。

解析：二叉树如下



4. 设有如下权值序列：W={7, 19, 2, 6, 32, 3, 21, 10}，求该权值序列的最优二叉树。

解析：最优二叉树如下



5. 设森林 F 中有 n 个非终端结点，用孩子兄弟算法表示该森林后得到一棵二叉树，试问该二叉树中有多少个右指针为空的结点？

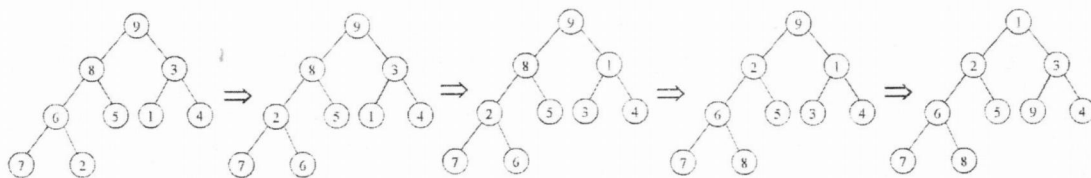
解析：有 $n+1$ 个右指针为空的结点；每个非终端的结点转换成二叉树后都对应于一个无右孩子的结点，最后一棵树的根结点也无右孩子，所以为 $n+1$ 个。

6. 设有字符串 S='a a b a b a a b a d'，求 Next[j]。

解析：Next[j]=0 1 1 1 2 1 2 3 4 5；参看课本 P82-P84。

7. 设关键字序列为{9,8,3,6, 5,1,4,7,2}，按初始堆创建算法将该序列调整为堆（首元素最小）。

解析：



（调整成小根堆）

8. 设有如下特殊矩阵 A，将其压缩存储到一维数组 SA 中

$$A = [i_3 j] \begin{pmatrix} a_{11}a_{12} & & & & & \\ & a_{22}a_{23} & & & & \\ & & a_{33}a_{34} & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & a_{n-1} \quad n-1 a_{n-1} \quad n \\ & & & & & & a_{nn} \end{pmatrix}$$

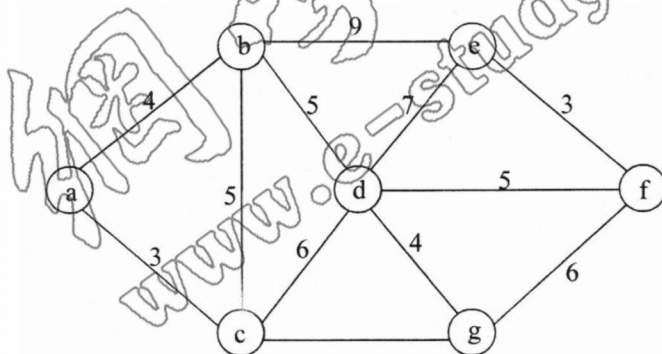
$$SA = a_{11}a_{12}a_{22}a_{23} \dots a_{n-1} \quad n-1 a_{n-1} \quad n, an, n$$

$$[k] \quad k=1 \dots 2n-1$$

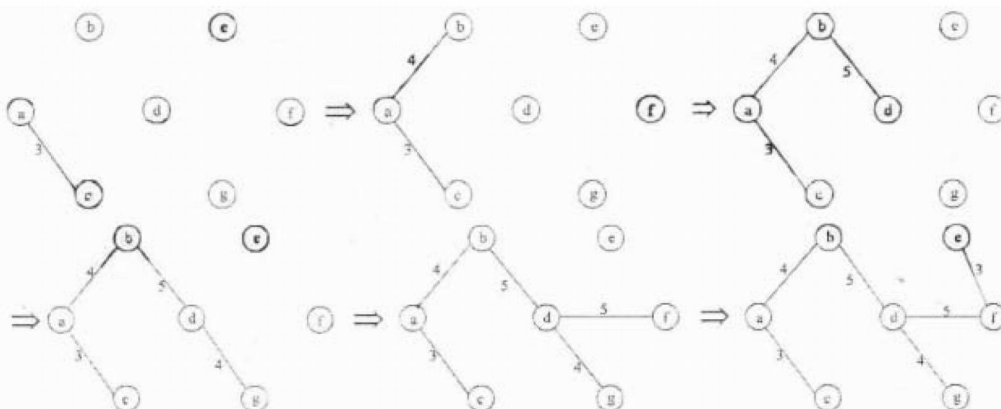
写出由下标[i,j]求 k 得转换方式。

解析： $k=i+j-1$ ； $k = \begin{cases} i+j-1, i=j \\ i+j-1, i=j-1 \end{cases}$

9. 下面为一个图，用普里姆算法求该图的最小生成树。



解析：prim 算法求最小生成树，过程如下：



10. 假设哈希空间为 0...12，哈希函数 $H(\text{key}) = (\text{key 的首字母在字母表中的序号} \bmod 13)$
 （注：A 字母序号为 0）；用开放地址法的线性再散列求如下关键字序列的哈希表。

(ZHAO, QIAN, SUN, LI, ZHOU, WU, ZHENG, WANG, ZHANG, CHAO, YANG)

解析：哈希表如下：

0	1	2	3	4	5	6	7	8	9	10	11	12
ZHO	ZHEN	ZHAN	QIA	CHA	SU	YAN			W	WAN	L	ZHA
U	G	G	N	O	N	G			U	G	I	O
2	3	4	1	3	1	9			1	2	1	1

三、阅读程序与填空（20 分）

1. 阅读下面程序，给出输出结果_____。

```
void demonstrate(){
    strassign(s, 'THIS A BOOK' );
    replace(s, substring(s, 3, 7), 'ESE, ARE' );
    concat(s, 'S' );
    printf( 's=' ,s);
} //demonstrate
```

解析：输出结果为：s= 'THESE ARE BOOK'；

SubString(&Sub, S, pos, len)用 Sub 返回串 s 的第 pos 个字符起长度为 len 的子串

Concat(&T, S1, S2)指用 T 返回 S1 和 S2 连接而成的新串

Replace(&s, T, V)指用 V 替换主串 s 中出现的所有与 T 相等的非重叠的子串

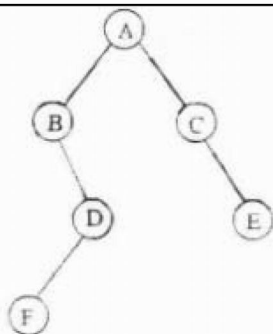
2. 设队列 Q 中的元素序列为：(1,3,5,7,9,2,4,6,8)；调用下面的算法后，队列 Q 中的元素序列改为：_____。

```
void exp(Queue &Q) {
    Stack S; int d;
    initStack(S);
    While(!queueempty(Q))
        {dequeue(Q, d); push(S, d); }
    While(!Stackempty(S))
        {pop(S, d); enqueue(Q, d); }
} //exp
```

解析：{8, 6, 4, 2, 9, 7, 5, 3, 1}；

该算法通过栈将元素逆置，栈的特点是‘先进后出’。这里先入栈后出栈，将各元素逆置再入队列，而队列是‘先进后出’，然后通过队列将元素输出。

3. 下图为一棵二叉树，阅读下面三个程序，给出访问二叉树的结果。



```

void visit1(T)
{if T
 {printf(T->data);
  visitl(T->rchild);
 Visitl(T->lchild);

```

```

vpid visit2(T)
{if T
 {visitl(T->rchild);
  printf(T->data);
  visitl(T->lchild);

```

```

vpid visit3(T)
{if T
 {visitl(T->rchild);
  visitl(T->lchild);
  printf(T->data);

```

运行结果：A C E B D F。 运行结果：ECADFB。 运行结果：ECFDBA。
 解析：改程序先访问根据点，再访问右孩子，最后访问左孩子，递归进行。

4. 下面为折半查找算法，填写适当的语句或条件，完善该算法。

```

iInt Search_Bin(SSTable ST, KeyType key);
{low=1; high=st.length;
  While(_____)
  {mid=_____/2;
   if EQ(key,STelem[mid].key)
    return mid;
   else if LT(key,STelem[mid].key)
    high=_____;
   else _____;
  }
  Return(0);
}

```

解析：**low<high**, **low+high**, **mid-1**, **low=mid+1**;
 这是一个折半查找算法，参看课本 P218

5、设有线性表 $L[1...8]=\{60,50,10,90,70,30,80,20\}$ ，调用 $\text{partition}(L,1,8)$ 之后，

$L[1...8]=\{\underline{\hspace{2cm}}\}$

函数值= ?

```

int partition(Splist&L, int low, int high)

```

```

{ temp=L.r[low];

```

```

while[low<high]

```

```

{ while(low<high)&&L.r[high].key>=temp.key--high;

```

```

  L.r[low]=L.r[high];

```

```

  while(low<high&&L.r[low].key<=temp.key)++low;

```

```

  L.r[high]=L.r[low];

```

```

}

```

```
L.r[low]=temp; return low;
}
```

解析：20,50,10, 30,60,70,80,90; 5

一趟快速排序算法，参看课本 P275

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

四、算法设计（28 分）

1.编写一个递归算法，计算二叉树中叶子结点的数目。

解析：

```
leaf_Nun(BiTree T)
{if(T==NULL)
return 0;
else if(T->lchild==NULL&&T->rchild==NULL)
return 1;
else
return(leaf_Nun(T->lchild)+leaf_Nun(T->rchild);
}
```

2.用递归方法改写折半查找算法（假设序列为整数序列，并存储在顺序存储线性表中）。

解析：

```
int search_Bin(SSTable ST,int low,int high,int key)
{int low=1,high=ST.Length;
int mid;
if(low<=high)
{mid=(low+high)/2;
if(ST[mid].key<k)
return search_Bin(ST,mid+1,high,k);
else if(ST[mid].key>k)
return search_Bin(ST,low,mid-1,k);
else return mid;
}
}
```

3.对 n ($n>2$) 个不同整型数据组成的序列，设计一算法，找出最大和最小的两个数据，要求比较次数少于 $2n-3$ （假设序列存储在顺序存储线性表中）。

解析：

```
int find(sqList&A, int&min, int&max)
{int min=A[0].key, max=A[0].key;
for(i=1;i<=n;i++)
{if(min>A[i].key)min=A[i].key;
else if(max<A[i].key)max=A[i].key;
else return min,max;
}
}
```

4. 已知一个线性表用来存储学生的成绩，试编写一个算法将成绩大于等于 60 分的置于线性表的前端，小于 60 分的放在线性表的后端。要求：

- (1) 采用顺序存储结构，除少量几个变量外不能利用附加的线性表，
- (2) 算法的时间复杂度为 $O(n)$ 。

解析：

```
void move(slist&L)
{ElemType temp;
int i=0,j=L.Length-1;
while(i<j)
{while(i<j && L.data[i]>=60) i++;
while(i<j && L.data[j]<60) j--;
if(i<j)
{temp=L.data[i];
L.data[i]=L.data[j];
L.data[j]=temp;
}
}
}
```

五、综合分析与设计（10 分）

设有 n 个星体，随着时间的变迁，相对于太阳的位置在不断的变化；为了跟踪各个星体的轨迹，每天都要测算各个星体相对于太阳的立体坐标（太阳坐标为 $(0,0,0)$ ），并存储在计算机中，要求始终保持最近一年的数据；同时要求能进行一些基本数据处理。

- (1) 分析数据，给出数据元素的存储结构描述；
- (2) 采用什么样的存储结构存储所有数据（给出存储结构的逻辑示意图）？为什么选择这种存储结构？
- (3) 设计一算法，求任意两星体间的距离。

解析：1)

```
typedef struct
{int a[n];
int addr[i][i][k];
}
```

2) 线性表的顺序示意图

存储地址	内存状态	元素在线表的位序
b	a_1	1
$b+1$	a_2	2
\vdots	\vdots	\vdots
$b+(i-1)l$	a_i	i
\vdots	\vdots	\vdots
$b+(i-1)l$	a_n	n

采用顺序存储结构可以随机访问任意一个位置的数据元素。

3)

```
#define      n      5000
int distance(int addr[][][])
```

```
{
    int a[n],i,j,k;
    int addr[M][N][P];
    int addr[I][J][K];
    int addr[0][0][0]=0;
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            for(k=0;k<n;k++)
            {
                addr[i][j][k]=sprt(I-M)^2+(J-N)^2+(K-P)^2);
            }
        }
    }
    for(i=0;i<n;i++)
    {
        for(j=0;j<n;j++)
        {
            for(k=0;k<n;k++)
            {
                printf( "%f" ,addr[i][j][k]);
                Printf( "/n" );
            }
        }
    }
}
```