

# 西南交通大学 2002 年硕士研究生招生入学考试

## 数据结构

## 试题

考试时间: 2002 年 1 月

考生请注意:

1. 本试题共 五 题, 共 六 页, 考生请认真检查;
2. 答题时, 直接将答题内容写在试题卷上;
3. 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										

### 选择与填空 (12 分)

1) 数据元素与数据对象  
数据对象 用于描述数据对象及数据元素间的关系; 而 数据结构 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。  
 2Ps: a) 数据 b) 数据结构 c) 数据类型 d) 存储结构

2) 二叉树第 K 层上至多有  $2^{K-1}$  个节点。  
 a)  $2K-1$  b)  $2^{K-1}$  c)  $2^K-1$

3) 下面算法的时间复杂度为:  $O(n^3)$ 。  

```

for (i=1; i<=n; i++) n+1
  for (j=1; j<=n; j++) n(n+1)
    {
      k=1;
      while (k<=n) k=5*k n^2
    }
    
```

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 \_\_\_\_\_;
- b) 在 P 结点前插入 S 结点的语句序列是 \_\_\_\_\_;
- c) 删除 P 结点的直接后继结点的语句序列是 \_\_\_\_\_;
- d) 删除 P 结点的直接前驱结点的语句序列是 \_\_\_\_\_;

报考专业:

姓名:

考生编号:

请不要在虚线内答题



- (5) S->next = P;  
 (6) S->priou = P;  
 (7) S->next = P;  
 (8) S->priou = P->priou;  
 (9) P->priou->next = P->next;  
 (10) P->priou->next = P;  
 (11) P->next->priou = P;  
 (12) P->next->priou = S;  
 (13) P->priou->next = S;  
 (14) P->next->priou = P->priou;  
 (15) Q = P->next;  
 (16) Q = P->priou;  
 (17) free(P);  
 (18) free(Q);

5) 要从 1000 个数据元素中选五个最小的, 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;

6)  $T(n) = O(f(n))$  中, 函数  $O(\quad)$  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  $O(n)$ 。

求解下面各问题 (30 分)

1) 已知有实现同一功能的两个算法, 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基本操作  $10^6$  次, 试问在此条件下, 这两个算法可解决问题的规模 (即  $n$  值的最大值) 各为多少? 哪个算法更适宜?  $O(n^{10})$  更适宜

$$O(10^n): 14 \quad O(n^{10}): 20$$

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。  

$$n = n_k + n_0$$
  

$$n_0 = (k-1)n_k + 1$$
  

$$n = 1 + k \cdot n_k$$

3) 假设一棵二叉数的先序序列为 EBADCFHGJK, 中序序列为 ABCDEFGHIJK, 请画出该二叉树。

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

5) 设森林  $F$  中有  $n$  个非终端结点, 用孩子兄弟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?

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

0 1 1 1 2 1 2 3 4 5

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).  $[\frac{9-1}{2}] = 4$

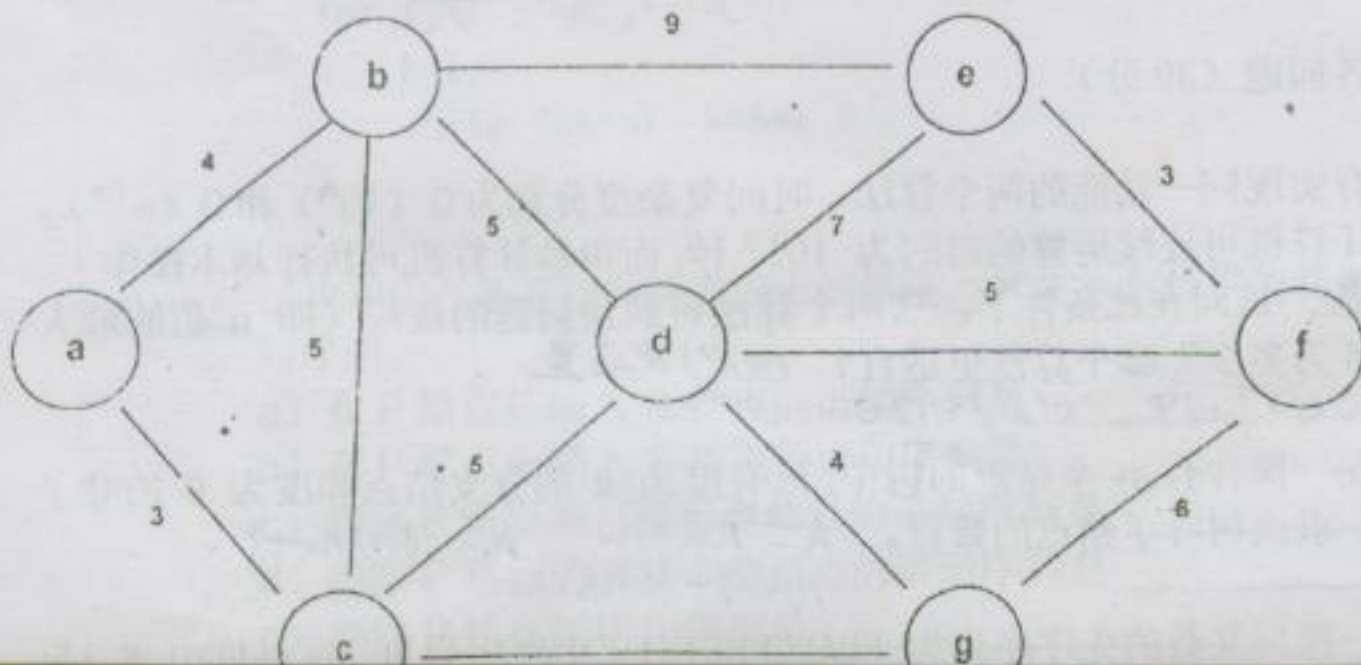
8) 设有如下特殊矩阵  $A$ , 将其压缩存储到一维数组  $SA$  中

$$A = \begin{pmatrix} a_{11} & a_{12} & & & & & & & \\ & a_{22} & a_{23} & & & & & & \\ & & a_{33} & a_{34} & & & & & \\ & & & & \ddots & & & & \\ & & & & & \ddots & & & \\ & & & & & & a_{n-1, n-1} & a_{n-1, n} \\ & & & & & & & a_n & n \end{pmatrix}$$

$$SA = \begin{matrix} a_{11} & a_{12} & a_{22} & a_{23} & \dots & a_{n-1, n-1} & a_{n-1, n} & a_{n, n} \\ [k] & k=1 \dots 2n-1 \end{matrix}$$

写出由下标  $[i, j]$  求  $k$  的转换公式。

9) 下面为一个图, 用普里姆算法求该图的最小生成树。





考生编号:

姓名:

报考专业:

请不要在虚线内答题

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}

阅读程序与填空 (20 分)

阅读下面程序, 给出输出结果

```
void demonstrate() {
```

```
    strassign(s, 'THIS IS A BOOK');
```

```
    replace(s, substring(s, 3, 7), 'ESE ARE'); → replace
```

```
    concat(s, 'S');
```

```
    printf('s=', s);
```

```
} //demonstrate
```

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

```
void exp(Queue &Q) {
```

```
    Stack S; int d;
```

```
    initstack(S);
```

```
    while (!queueempty(Q))
```

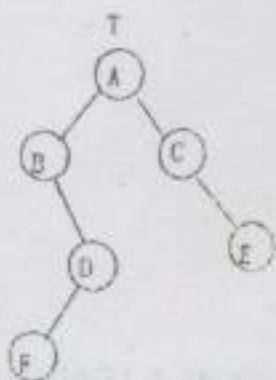
```
    { dequeue(Q, d); push(S, d); }
```

```
    while (!stackempty(S))
```

```
    { pop(S, d); enqueue(Q, d); }
```

```
//exp
```

> 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);
```

```
      visit1(T->rchild);
```

```
      visit1(T->lchild);
```

```
    }
```

```
} 递归
```

运行结果: ABDFCE

```
Void visit2(T) {
```

```
Void visit3(T) {
```



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。

```

Int Search_Bin(SSTable ST, KeyType key):
{
    low=1; high=st.length;
    while ( low < high )
    {
        mid= (low+high)/2;
        if EQ(key, ST.elem[mid].key)
            return mid;
        else if LT(key, ST.elem[mid].key)
            high = mid-1;
        else low=mid+1;
    }
    return(0);
}

```

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

之后,  $L[1..8]=\{ \underline{20, 50, 10, 30, 60, 70, 80, 90} \}$ ;

函数值= 5 ?

```

int partition(Sqlist &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;
}

```

(\*\*\*\*\*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。
- 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)。
- 3) 对  $n$  ( $n>2$ ) 个不同整形数据组成的序列, 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)。



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) 时间复杂度为  $O(n)$ 。

- (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  $O(n)$ 。

### 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（太阳坐标恒为

$(0,0,0)$ ），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。

```

2) int search-bin (SSTable ST, keyType key)
{
    low = 1; high = ST.length;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (ST.elem[mid] == key)
            return mid;
        else if (ST.elem[mid] < key)
            low = mid + 1;
        else
            high = mid - 1;
    }
}

```

```

2) int Search-Bin (SSTable ST, int key, int low, int high)
{
    if (low > high) return 0;
    mid = (low + high) / 2;
    if (ST.elem[mid] == key) return mid;
    else if (ST.elem[mid] < key)
        return Search-Bin (ST, key, mid + 1, high);
    else
        return Search-Bin (ST, key, low, mid - 1);
}

```

```

3) find (SgList &L, &min, &max)
{
    min = L.RZ[1].key;
    max = L.RZ[1].key;
    for (i = 1; i <= L.length; i++)
    {
        if (L.RZ[i].key < min)
            min = L.RZ[i].key;
        else if (L.RZ[i].key > max)
            max = L.RZ[i].key;
    }
}

```

```

4) 算法实现从有序表中删除重复元素
Order (SgList &L, int n)
{
    int i = 1, j = 1;
    while (i <= n)
    {
        while (i < j && L.RZ[i].key == L.RZ[j].key) j++;
        while (i < j && L.RZ[j].key < L.RZ[i].key) j--;
        L.RZ[i].key = L.RZ[j].key;
        i++;
    }
}
4) Order (SgList &L, int n)
{
    int i = 1, j = 1;
    while (i <= n)
    {
        while (i < j && L.elem[i] == L.elem[j]) j++;
        while (i < j && L.elem[j] < L.elem[i]) j--;
        if (i < j)
            L.elem[i] = L.elem[j];
        i++;
    }
}

```