



书彦教育是一家考研专业课培训机构，专注于收集各高校的考研咨询，考研辅导课程，考研图书，复习资料、历年考研真题，考研名师视频，模拟密押卷、冲刺预测卷、考研公共课……

书彦教育官网在每个大学均有十余名研究生负责维护，各个专业的资料都来源于本校内部，所有真题都是本校研究生在研招办或其他途径收集购买，书彦教育专业学术团队负责把关，确保考生拿到手的每一份资料都是报考院校最有价值的考研资料，与文都考研长期合作，为广大考生提供最全面最有价值考研资料，最及时最可靠的考研信息。经过十年的发展，书彦教育已经成为中国考研专业课最权威教育机构之一。

## 华南师范大学 2015 年攻读硕士学位研究生入学考试试题

考试科目：数据结构      科目代码：925      考试时间：    月    日

(注：特别提醒所有答案一律写在答题纸上，直接写在试题或草稿纸上的无效!)

### 一、选择题

1. 设一组权值集合  $W=\{2, 3, 4, 5, 6\}$ ，则由该权值集合构造的哈夫曼树中带权路径长度之和为 ( D )。  
A、20      B、30      C、40      D、45
2. 执行一趟快速排序能够得到的序列是 ( A )。  
A、 $[41, 12, 34, 45, 27] \ 55 \ [72, 63]$   
B、 $[45, 34, 12, 41] \ 55 \ [72, 63, 27]$   
C、 $[63, 12, 34, 45, 27] \ 55 \ [41, 72]$   
D、 $[12, 27, 45, 41] \ 55 \ [34, 63, 72]$
3. 设一条单链表的头指针变量为 head 且该链表没有头结点，则其判空条件是 ( A )。  
A、 $head==0$       B、 $head \rightarrow next==0$   
C、 $head \rightarrow next==head$       D、 $head!=0$
4. 时间复杂度不受数据初始状态影响而恒为  $O(n \log_2 n)$  的是 ( A )。  
A、堆排序      B、冒泡排序      C、希尔排序      D、快速排序
5. 设二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树满足的条件是 ( D )。  
A、空或只有一个结点      B、高度等于其结点数  
C、任一结点无左孩子      D、任一结点无右孩子
6. 一趟排序结束后不一定能够选出一个元素放在其最终位置上的是 ( D )。  
A、堆排序      B、冒泡排序      C、快速排序      D、希尔排序
7. 设某棵二叉树中有 40 个结点，则该二叉树的最小高度为 ( B )。  
A、3      B、4      C、5      D、6

8. 顺序查找不论在顺序线性表中还是在链式线性表中的时间复杂度为 ( A )。  
A、  $O(n)$                       B、  $O(n^2)$                       C、  $O(n^{1/2})$                       D、  $O(\log_2 n)$
9. 二路归并排序的时间复杂度为 ( C )。  
A、  $O(n)$                       B、  $O(n^2)$                       C、  $O(n \log_2 n)$                       D、  $O(\log_2 n)$
10. 深度为 k 的完全二叉树中最少有 ( B ) 个结点。  
A、  $2^{k-1}-1$                       B、  $2^{k-1}$                       C、  $2^{k-1}+1$                       D、  $2^k-1$
11. 设指针变量 front 表示链式队列的队头指针, 指针变量 rear 表示链式队列的队尾指针, 指针变量 s 指向将要入队列的结点 X, 则入队列的操作序列为 ( C )。  
A、 front->next=s; front=s;                      B、 s->next=rear; rear=s;  
C、 rear->next=s; rear=s;                      D、 s->next=front; front=s;
12. 设某无向图中有 n 个顶点 e 条边, 则建立该图邻接表的时间复杂度为 ( A )。  
A、  $O(n+e)$                       B、  $O(n^2)$                       C、  $O(ne)$                       D、  $O(n^3)$
13. 设某哈夫曼树中有 199 个结点, 则该哈夫曼树中有 ( B ) 个叶子结点。  
A、 99                      B、 100                      C、 101                      D、 102
14. 设二叉排序树上有 n 个结点, 则在二叉排序树上查找结点的平均时间复杂度为 ( D )。  
A、  $O(n)$                       B、  $O(n^2)$                       C、  $O(n \log_2 n)$                       D、  $O(\log_2 n)$
15. 设用邻接矩阵 A 表示有向图 G 的存储结构, 则有向图 G 中顶点 i 的入度为 ( B )。  
A、 第 i 行非 0 元素的个数之和                      B、 第 i 列非 0 元素的个数之和  
C、 第 i 行 0 元素的个数之和                      D、 第 i 列 0 元素的个数之和

## 二、判断题

1. 调用一次深度优先遍历可以访问到图中的所有顶点。(错)
2. 分块查找的平均查找长度不仅与索引表的长度有关, 而且与块的长度有关。(对)
3. 冒泡排序在初始关键字序列为逆序的情况下执行的交换次数最多。(对)
4. 满二叉树一定是完全二叉树, 完全二叉树不一定是满二叉树。(对)
5. 设一棵二叉树的先序序列和后序序列, 则能够唯一确定出该二叉树的形状。(错)
6. 层次遍历初始堆可以得到一个有序的序列。(错)
7. 设一棵树 T 可以转化成二叉树 BT, 则二叉树 BT 中一定没有右子树。(对)
8. 线性表的顺序存储结构比链式存储结构更好。(错)
9. 中序遍历二叉排序树可以得到一个有序的序列。(对)
10. 快速排序是排序算法中平均性能最好的一种排序。(对)

## 三、填空题

1. for(i=1, t=1, s=0; i<=n; i++) {t=t\*i; s=s+t; } 的时间复杂度为 (  $O(n)$  )。
2. 设指针变量 p 指向单链表中结点 A, 指针变量 s 指向被插入的新结点 X, 则进行插入操作的语句序列为 ( s->next=p->next; p->next=s ) (设结点的指针域为 next)。
3. 设有向图 G 的二元组形式表示为  $G = (D, R)$ ,  $D = \{1, 2, 3, 4, 5\}$ ,  $R = \{r\}$ ,  $r = \{<1,2>, <2,4>, <4,5>, <1,3>, <3,2>, <3,5>\}$ , 则给出该图的一种拓扑排序序列 ( (1, 3, 2, 4, 5) )。
4. 设无向图 G 中有 n 个顶点, 则该无向图中每个顶点的度数最多是 ( n-1 )。
5. 设二叉树中度数为 0 的结点数为 50, 度数为 1 的结点数为 30, 则该二叉树中总共有 (129) 个结点数。
6. 设 F 和 R 分别表示顺序循环队列的头指针和尾指针, 则判断该循环队列为空的条件为 ( F==R )。

7. 设二叉树中结点的两个指针域分别为 lchild 和 rchild, 则判断指针变量 p 所指向的结点为叶子结点的条件是 (  $p \rightarrow lchild == 0 \&\& p \rightarrow rchild == 0$  )。
8. 简单选择排序和直接插入排序算法的平均时间复杂度为 (  $O(n^2)$  )。
9. 快速排序算法的空间复杂度平均情况下为 (  $O(n \log_2 n)$  ), 最坏的情况下为 (  $O(n)$  )。
10. 散列表中解决冲突的两种方法是 ( 开放定址法 ) 和 ( 链地址法 )。

#### 四、算法设计题

1. 设计在顺序有序表中实现二分查找的算法。

参考答案:

```
struct record {int key; int others;};
int bisearch(struct record r[ ], int k)
{
    int low=0, mid, high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(r[mid].key==k) return(mid+1); else if(r[mid].key>k) high=mid-1; else low=mid+1;
    }
    return(0);
}
```

2. 设计判断二叉树是否为二叉排序树的算法。

参考答案:

```
int minnum=-32768, flag=1;
typedef struct node {int key; struct node *lchild, *rchild;} bitree;
void inorder(bitree *bt)
{
    if(bt!=0) {inorder(bt->lchild); if(minnum>bt->key) flag=0; minnum=bt->key; inorder(bt->rchild);}
}
```

3. 在链式存储结构上设计直接插入排序算法

参考答案:

```
void straightinsertsort(lklist *&head)
{
    lklist *s, *p, *q; int t;
    if (head==0 || head->next==0) return;
    else for(q=head, p=head->next; p!=0; p=q->next)
    {
        for(s=head; s!=q->next; s=s->next) if (s->data>p->data) break;
        if(s==q->next) q=p;
    }
    else {q->next=p->next; p->next=s->next; s->next=p; t=p->data; p->data=s->data; s->data=t;}
}
```

## 华南师范大学 2014 年攻读硕士学位研究生入学考试试题

考试科目：数据结构 科目代码：925 考试时间： 月 日

(注：特别提醒所有答案一律写在答题纸上，直接写在试题或草稿纸上的无效!)

## 一、选择题

1. 设某无向图有  $n$  个顶点，则该无向图的邻接表中有 ( B ) 个表头结点。  
A、  $2n$                       B、  $n$                       C、  $n/2$                       D、  $n(n-1)$
2. 设无向图  $G$  中有  $n$  个顶点，则该无向图的最小生成树上有 ( B ) 条边。  
A、  $n$                       B、  $n-1$                       C、  $2n$                       D、  $2n-1$
3. 设一组初始记录关键字序列为 (60, 80, 55, 40, 42, 85)，则以第一个关键字 45 为基准而得到的一趟快速排序结果是 ( C )。  
A、 40, 42, 60, 55, 80, 85                      B、 42, 45, 55, 60, 85, 80  
C、 42, 40, 55, 60, 80, 85                      D、 42, 40, 60, 85, 55, 80
4. ( B ) 二叉排序树可以得到一个从小到大的有序序列。  
A、 先序遍历                      B、 中序遍历                      C、 后序遍历                      D、 层次遍历
5. 设按照从上到下、从左到右的顺序从 1 开始对完全二叉树进行顺序编号，则编号为  $i$  结点的左孩子结点的编号为 ( B )。  
A、  $2i+1$                       B、  $2i$                       C、  $i/2$                       D、  $2i-1$
6. 程序段  $s=i=0$ ; **do** { $i=i+1$ ;  $s=s+i$ ; } **while**( $i \leq n$ ); 的时间复杂度为 ( A )。  
A、  $O(n)$                       B、  $O(n \log_2 n)$                       C、  $O(n^2)$                       D、  $O(n^3/2)$
7. 设带有头结点的单向循环链表的头指针变量为  $head$ ，则其判空条件是 ( C )。  
A、  $head==0$                       B、  $head \rightarrow next==0$   
C、  $head \rightarrow next==head$                       D、  $head!=0$
8. 设某棵二叉树的高度为 10，则该二叉树上叶子结点最多有 ( C )。  
A、 20                      B、 256                      C、 512                      D、 1024
9. 设一组初始记录关键字序列为 (13, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134)，则利用二分法查找关键字 90 需要比较的关键字个数为 ( B )。  
A、 1                      B、 2                      C、 3                      D、 4
10. 设指针变量  $top$  指向当前链式栈的栈顶，则删除栈顶元素的操作序列为 ( D )。  
A、  $top=top+1$ ;                      B、  $top=top-1$ ;  
C、  $top \rightarrow next=top$ ;                      D、  $top=top \rightarrow next$ ;

## 二、判断题

1. 不论是入队列操作还是入栈操作，在顺序存储结构上都需要考虑“溢出”情况。(对)
2. 当向二叉排序树中插入一个结点，则该结点一定成为叶子结点。(对 )
3. 设某堆中有  $n$  个结点，则在该堆中插入一个新结点的时间复杂度为  $O(\log_2 n)$ 。(对 )
4. 完全二叉树中的叶子结点只可能在最后两层中出现。(对)
5. 哈夫曼树中没有度数为 1 的结点。(对)

6. 对连通图进行深度优先遍历可以访问到该图中的所有顶点。(对)
7. 先序遍历一棵二叉排序树得到的结点序列不一定是有序的序列。(对)
8. 由树转化成二叉树, 该二叉树的右子树不一定为空。(错)
9. 线性表中的所有元素都有一个前驱元素和后继元素。(错)
10. 带权无向图的最小生成树是唯一的。(错)

### 三、填空题

1. 设指针变量 p 指向双向链表中的结点 A, 指针变量 s 指向被插入的结点 X, 则在结点 A 的后面插入结点 X 的操作序列为 (s->left=p) =p; s->right=p->right; (p->right) =s; p->right->left=s; (设结点中的两个指针域分别为 left 和 right)。
2. 设完全有向图中有 n 个顶点, 则该完全有向图中共有 (n(n-1)) 条有向边; 设完全无向图中有 n 个顶点, 则该完全无向图中共有 (n(n-1)/2) 条无向边。
3. 设关键字序列为 (K<sub>1</sub>, K<sub>2</sub>, ..., K<sub>n</sub>), 则用筛选法建初始堆必须从第 (n/2) 个元素开始进行筛选。
4. 解决散列表冲突的两种方法是 (开放定址法) 和 (链地址法)。
5. 设一棵三叉树中有 50 个度数为 0 的结点, 21 个度数为 2 的结点, 则该二叉树中度数为 3 的结点数有 (14) 个。
6. 高度为 h 的完全二叉树中最少有 (2<sup>h-1</sup>) 个结点, 最多有 (2<sup>h</sup>-1) 个结点。
7. 设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟直接插入排序结束后的结果的是 (12, 24, 35, 27, 18, 26)。
8. 设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟简单选择排序结束后的结果的是 (12, 18, 24, 27, 35, 26)。
9. 设一棵二叉树的前序序列为 ABC, 则有 (5) 种不同的二叉树可以得到这种序列。
10. 下面程序段的功能是实现一趟快速排序, 请在下划线处填上正确的语句。

```

struct record {int key;datatype others;};
void quickpass(struct record r[], int s, int t, int &i)
{
    int j=t; struct record x=r[s]; i=s;
    while(i<j)
    {
        while (i<j && r[j].key>x.key) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}
        while ( (i<j && r[i].key<x.key) ) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}
    }
    ( r[i]=x );
}

```

### 四、算法设计题

1. 设计在链式结构上实现简单选择排序算法。

参考答案:

```

void simpleselectsort(lklist *&head)
{
    lklist *p,*q,*s; int min,t;
    if(head==0 || head->next==0) return;

```

```

for(q=head; q!=0;q=q->next)
{
    min=q->data; s=q;
    for(p=q->next; p!=0;p=p->next) if(min>p->data){min=p->data; s=p;}
    if(s!=q){t=s->data; s->data=q->data; q->data=t;}
}
}

```

2. 设计在顺序存储结构上实现求子串算法。

**参考答案:**

```

void substring(char s[ ], long start, long count, char t[ ])
{
    long i,j,length=strlen(s);
    if (start<1 || start>length) printf("The copy position is wrong");
    else if (start+count-1>length) printf("Too characters to be copied");
    else { for(i=start-1,j=0; i<start+count-1;i++,j++) t[j]=s[i]; t[j]='\0';}
}

```

3. 设计求结点在二叉排序树中层次的算法。

**参考答案:**

```

int lev=0;
typedef struct node{int key; struct node *lchild,*rchild;} bitree;
void level(bitree *bt,int x)
{
    if (bt!=0)
    {lev++; if (bt->key==x) return; else if (bt->key>x) level(bt->lchild,x); else level(bt->rchild,x);}
}

```



## 华南师范大学 2013 年攻读硕士学位研究生入学考试试题

考试科目：数据结构 科目代码：925 考试时间： 月 日

(注：特别提醒所有答案一律写在答题纸上，直接写在试题或草稿纸上的无效!)

## 一、选择题

1. 字符串的长度是指 ( C )。  
A、串中不同字符的个数 B、串中不同字母的个数  
C、串中所含字符的个数 D、串中不同数字的个数
2. 建立一个长度为  $n$  的有序单链表的时间复杂度为 ( C )  
A、 $O(n)$  B、 $O(1)$  C、 $O(n^2)$  D、 $O(\log_2 n)$
3. 两个字符串相等的充要条件是 ( C )。  
A、两个字符串的长度相等 B、两个字符串中对应位置上的字符相等  
C、同时具备(A)和(B)两个条件 D、以上答案都不对
4. 设某散列表的长度为 100，散列函数  $H(k)=k \% P$ ，则  $P$  通常情况下最好选择 ( B )。  
A、99 B、97 C、91 D、93
5. 在二叉排序树中插入一个关键字值的平均时间复杂度为 ( B )。  
A、 $O(n)$  B、 $O(\log_2 n)$  C、 $O(n \log_2 n)$  D、 $O(n^2)$
6. 设一个顺序有序表  $A[1:14]$  中有 14 个元素，则采用二分法查找元素  $A[4]$  的过程中比较元素的顺序为 ( C )。  
A、 $A[1], A[2], A[3], A[4]$  B、 $A[1], A[14], A[7], A[4]$   
C、 $A[7], A[3], A[5], A[4]$  D、 $A[7], A[5], A[3], A[4]$
7. 设一棵完全二叉树中有 65 个结点，则该完全二叉树的深度为 ( B )。  
A、8 B、7 C、6 D、5
8. 设一棵三叉树中有 2 个度数为 1 的结点，2 个度数为 2 的结点，2 个度数为 3 的结点，则该三叉链权中有 ( C ) 个度数为 0 的结点。  
A、5 B、6 C、7 D、8
9. 设无向图  $G$  中的边的集合  $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点  $a$  出发进行深度优先遍历可以得到的一种顶点序列为 ( A )。  
A、aedfcb B、acfebd C、aebcfd D、aedfbc
10. 队列是一种 ( A ) 的线性表。  
A、先进先出 B、先进后出 C、只能插入 D、只能删除

## 二、判断题

1. 如果两个关键字的值不等但哈希函数值相等，则称这两个关键字为同义词。(对)
2. 设初始记录关键字基本有序，则快速排序算法的时间复杂度为  $O(n \log_2 n)$ 。(错)
3. 分块查找的基本思想是首先在索引表中进行查找，以便确定给定的关键字可能存在的块号，然后再在相应的块内进行顺序查找。(对)

4. 二维数组和多维数组均不是特殊的线性结构。(错)
5. 向二叉排序树中插入一个结点需要比较的次数可能大于该二叉树的高度。(错)
6. 如果某个有向图的邻接表中第  $i$  条单链表为空, 则第  $i$  个顶点的出度为零。(对)
7. 非空的双向循环链表中任何结点的前驱指针均不为空。(对)
8. 不论线性表采用顺序存储结构还是链式存储结构, 删除值为  $X$  的结点的时间复杂度均为  $O(n)$ 。(对)
9. 图的深度优先遍历算法中需要设置一个标志数组, 以便区分图中的每个顶点是否被访问过。(对)
10. 稀疏矩阵的压缩存储可以用一个三元组表来表示稀疏矩阵中的非 0 元素。(对)

### 三、填空题

1. 设一组初始记录关键字序列为(49, 38, 65, 97, 76, 13, 27, 50), 则以  $d=4$  为增量的一趟希尔排序结束后的结果为 (49, 13, 27, 50, 76, 38, 65, 97)。
2. 下面程序段的功能是实现在二叉排序树中插入一个新结点, 请在下划线处填上正确的内容。  

```
typedef struct node{int data;struct node *lchild;struct node *rchild;}bitree;
void bstinsert(bitree *&t,int k)
{
    if(t==0){      (      t=(bitree *)malloc(sizeof(bitree))      );
        t->data=k;t->lchild=t->rchild=0;}
    else if (t->data>k) bstinsert(t->lchild,k);else (      bstinsert(t->rchild,k)      );
}
```
3. 设指针变量  $p$  指向单链表中结点  $A$ , 指针变量  $s$  指向被插入的结点  $X$ , 则在结点  $A$  的后面插入结点  $X$  需要执行的语句序列:  $s->next=p->next$ ; ( $p->next=s$ );。
4. 设指针变量  $head$  指向双向链表中的头结点, 指针变量  $p$  指向双向链表中的第一个结点, 则指针变量  $p$  和指针变量  $head$  之间的关系是  $p=(head->rlink)$  和  $head=(p->llink)$  (设结点中的两个指针域分别为  $llink$  和  $rlink$ )。
5. 设某棵二叉树的中序遍历序列为  $ABCD$ , 后序遍历序列为  $BADC$ , 则其前序遍历序列为 ( $CABD$ )。
6. 完全二叉树中第 5 层上最少有 (1) 个结点, 最多有 (16) 个结点。
7. 设有向图中不存在有向边  $\langle V_i, V_j \rangle$ , 则其对应的邻接矩阵  $A$  中的数组元素  $A[i][j]$  的值等于 (0)。
8. 设一组初始记录关键字序列为(49, 38, 65, 97, 76, 13, 27, 50), 则第 4 趟直接选择排序结束后的结果为(13, 27, 38, 50, 76, 49, 65, 97)。
9. 设连通图  $G$  中有  $n$  个顶点  $e$  条边, 则对应的最小生成树上有  $(n-1)$  条边。
10. 设有一组初始记录关键字序列为(50, 16, 23, 68, 94, 70, 73), 则将它们调整成初始堆只需把 16 与 (50) 相互交换即可。

### 四、算法设计题

1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。

参考答案:

```
void countnode(bitree *bt,int &count)
{
    if(bt!=0)
    {count++; countnode(bt->lchild,count); countnode(bt->rchild,count);}
}
```



}

2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

**参考答案:**

```
typedef struct {int vertex[m]; int edge[m][m];}gadjmatrix;
typedef struct node1 {int info;int adjvertex; struct node1 *nextarc;} glinklistnode;
typedef struct node2 {int vertexinfo;glinklistnode *firstarc;} glinkheadnode;
void adjmatrixtoadjlist(gadjmatrix g1[ ][ ],glinkheadnode g2[ ])
{
    int i,j; glinklistnode *p;
    for(i=0;i<=n-1;i++) g2[i].firstarc=0;
    for(i=0;i<=n-1;i++) for(j=0;j<=n-1;j++)
        if (g1.edge[i][j]==1)
        {
            p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=j;
            p->nextarc=g[i].firstarc; g[i].firstarc=p;
            p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=i;
            p->nextarc=g[j].firstarc; g[j].firstarc=p;
        }
}
```

## 华南师范大学 2012 年攻读硕士学位研究生入学考试试题

考试科目：数据结构 科目代码：925 考试时间： 月 日

(注：特别提醒所有答案一律写在答题纸上，直接写在试题或草稿纸上的无效!)

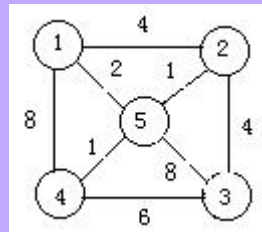
## 一、选择题

1. 下列程序段的时间复杂度为 ( A )。  
for(i=0; i<m; i++) for(j=0; j<t; j++) c[i][j]=0;  
for(i=0; i<m; i++) for(j=0; j<t; j++) for(k=0; k<n; k++) c[i][j]=c[i][j]+a[i][k]\*b[k][j];  
A、  $O(m*n*t)$  B、  $O(m+n+t)$  C、  $O(m+n*t)$  D、  $O(m*t+n)$
2. 设顺序线性表中有  $n$  个数据元素，则删除表中第  $i$  个元素需要移动 ( A ) 个元素。  
A、  $n-i$  B、  $n+1-i$  C、  $n-1-i$  D、  $i$
3. 设  $F$  是由  $T_1$ 、 $T_2$  和  $T_3$  三棵树组成的森林，与  $F$  对应的二叉树为  $B$ ， $T_1$ 、 $T_2$  和  $T_3$  的结点数分别为  $N_1$ 、 $N_2$  和  $N_3$ ，则二叉树  $B$  的根结点的左子树的结点数为 ( A )。  
A、  $N_1-1$  B、  $N_2-1$  C、  $N_2+N_3$  D、  $N_1+N_3$
4. 利用直接插入排序法的思想建立一个有序线性表的时间复杂度为 ( C )。  
A、  $O(n)$  B、  $O(n\log_2 n)$  C、  $O(n^2)$  D、  $O(\log_2 n)$
5. 设指针变量  $p$  指向双向链表中结点  $A$ ，指针变量  $s$  指向被插入的结点  $X$ ，则在结点  $A$  的后面插入结点  $X$  的操作序列为 ( D )。  
A、  $p \rightarrow \text{right} = s; s \rightarrow \text{left} = p; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{right} = p \rightarrow \text{right};$   
B、  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s;$   
C、  $p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right};$   
D、  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} \rightarrow \text{left} = s; p \rightarrow \text{right} = s;$
6. 下列各种排序算法中平均时间复杂度为  $O(n^2)$  是 ( D )。  
A、 快速排序 B、 堆排序 C、 归并排序 D、 冒泡排序
7. 设输入序列  $1、2、3、\dots、n$  经过栈作用后，输出序列中的第一个元素是  $n$ ，则输出序列中的第  $i$  个输出元素是 ( C )。  
A、  $n-i$  B、  $n-1-i$  C、  $n+1-i$  D、 不能确定
8. 设散列表中有  $m$  个存储单元，散列函数  $H(\text{key}) = \text{key} \% p$ ，则  $p$  最好选择 ( B )。  
A、 小于等于  $m$  的最大奇数 B、 小于等于  $m$  的最大素数  
C、 小于等于  $m$  的最大偶数 D、 小于等于  $m$  的最大合数
9. 设在一棵度数为 3 的树中，度数为 3 的结点数有 2 个，度数为 2 的结点数有 1 个，度数为 1 的结点数有 2 个，那么度数为 0 的结点数有 ( C ) 个。  
A、 4 B、 5 C、 6 D、 7
10. 设完全无向图中有  $n$  个顶点，则该完全无向图中有 ( A ) 条边。  
A、  $n(n-1)/2$  B、  $n(n-1)$  C、  $n(n+1)/2$  D、  $(n-1)/2$
11. 设顺序表的长度为  $n$ ，则顺序查找的平均比较次数为 ( C )。  
A、  $n$  B、  $n/2$  C、  $(n+1)/2$  D、  $(n-1)/2$

12. 设有序表中的元素为(13, 18, 24, 35, 47, 50, 62), 则在其中利用二分法查找值为 24 的元素需要经过 ( C ) 次比较。  
A、 1                      B、 2                      C、 3                      D、 4
13. 设顺序线性表的长度为 30, 分成 5 块, 每块 6 个元素, 如果采用分块查找, 则其平均查找长度为 ( D )。  
A、 6                      B、 11                      C、 5                      D、 6.5
14. 设有向无环图 G 中的有向边集合  $E=\{<1, 2>, <2, 3>, <3, 4>, <1, 4>\}$ , 则下列属于该有向图 G 的一种拓扑排序序列的是 ( A )。  
A、 1, 2, 3, 4    B、 2, 3, 4, 1    C、 1, 4, 2, 3    D、 1, 2, 4, 3
15. 设有一组初始记录关键字序列为(34, 76, 45, 18, 26, 54, 92), 则由这组记录关键字生成的二叉排序树的深度为 ( A )。  
A、 4                      B、 5                      C、 6                      D、 7

## 二、填空题

- 设指针 p 指向单链表中结点 A, 指针 s 指向被插入的结点 X, 则在结点 A 的前面插入结点 X 时的操作序列为:  
1)  $s \rightarrow next = (p \rightarrow next)$ ; 2)  $p \rightarrow next = s$ ; 3)  $t = p \rightarrow data$ ;  
4)  $p \rightarrow data = (s \rightarrow data)$ ; 5)  $s \rightarrow data = t$ ;
- 设某棵完全二叉树中有 100 个结点, 则该二叉树中有 (50) 个叶子结点。
- 设某顺序循环队列中有 m 个元素, 且规定队头指针 F 指向队头元素的前一个位置, 队尾指针 R 指向队尾元素的当前位置, 则该循环队列中最多存储 (m-1) 队列元素。
- 对一组初始关键字序列 (40, 50, 95, 20, 15, 70, 60, 45, 10) 进行冒泡排序, 则第一趟需要进行相邻记录的比较的次数为 (6), 在整个排序过程中最多需要进行 (8) 趟排序才可以完成。
- 在堆排序和快速排序中, 如果从平均情况下排序的速度最快的角度来考虑应最好选择 (快速) 排序, 如果从节省存储空间的角度来考虑则最好选择 (堆) 排序。
- 设一棵二叉树的中序遍历序列为 BDCA, 后序遍历序列为 DBAC, 则这棵二叉树的前序序列为 ( CBDA )。
- 设用于通信的电文仅由 8 个字母组成, 字母在电文中出现的频率分别为 7、19、2、6、32、3、21、10, 根据这些频率作为权值构造哈夫曼树, 则这棵哈夫曼树的高度为 (6)。
- 设一组记录关键字序列为(80, 70, 33, 65, 24, 56, 48), 则用筛选法建成的初始堆为 (24, 65, 33, 80, 70, 56, 48)。
- 设无向图 G (如右图所示), 则其最小生成树上所有边的权值之和为 (8)。



10. 设一组初始记录关键字序列为(20, 12, 42, 31, 18, 14, 28), 则根据这些记录关键字构造的二叉排序树的平均查找长度是 (19/7)。

## 三、判断题

- 有向图的邻接表和逆邻接表中表结点的个数不一定相等。(错)
- 对链表进行插入和删除操作时不必移动链表中结点。(对)
- 子串“ABC”在主串“AABCABCD”中的位置为 2。(对)
- 若一个叶子结点是某二叉树的中序遍历序列的最后一个结点, 则它必是该二叉树的先序遍历序列中的最后一个结点。(对)

5. 希尔排序算法的时间复杂度为  $O(n^2)$ 。(错)
6. 用邻接矩阵作为图的存储结构时,则其所占用的存储空间与图中顶点数无关而与图中边数有关。(错)
7. 中序遍历一棵二叉排序树可以得到一个有序的序列。(对)
8. 入栈操作和入队列操作在链式存储结构上实现时不需要考虑栈溢出的情况。(对)
9. 顺序表查找指的是在顺序存储结构上进行查找。(错)
10. 堆是完全二叉树,完全二叉树不一定是堆。(对)

## 五、算法设计题

1. 设计计算二叉树中所有结点值之和的算法。

参考答案:

```
void sum(bitree *bt,int &s)
{
    if(bt!=0) {s=s+bt->data; sum(bt->lchild,s); sum(bt->rchild,s);}
}
```

2. 设计将所有奇数移到所有偶数之前的算法。

参考答案:

```
void quickpass(int r[], int s, int t)
{
    int i=s,j=t,x=r[s];
    while(i<j)
    {
        while (i<j && r[j]%2==0) j=j-1;  if (i<j) {r[i]=r[j];i=i+1;}
        while (i<j && r[i]%2==1) i=i+1;  if (i<j) {r[j]=r[i];j=j-1;}
    }
    r[i]=x;
}
```

3. 设计判断单链表中元素是否是递增的算法。

参考答案:

```
int isriselk(lklist *head)
{
    if(head==0||head->next==0) return(1);else
    for(q=head,p=head->next; p!=0; q=p,p=p->next)if(q->data>p->data) return(0);
    return(1);
}
```

## 华南师范大学 2016 年攻读硕士学位研究生入学考试试题

考试科目：数据结构 科目代码：925 考试时间： 月 日

(注：特别提醒所有答案一律写在答题纸上，直接写在试题或草稿纸上的无效!)

## 一、选择题

1. 下列程序段的时间复杂度为 ( A )。  
 $i=0, s=0; \text{ while } (s \leq n) \{ s=s+i; i++; \}$   
A、 $O(n^{1/2})$  B、 $O(n^{1/3})$  C、 $O(n)$  D、 $O(n^2)$
2. 设某链表中最常用的操作是在链表的尾部插入或删除元素，则选用下列 ( D ) 存储方式最节省运算时间。  
A、单向链表 B、单向循环链表 C、双向链表 D、双向循环链表
3. 设指针 q 指向单链表中结点 A，指针 p 指向单链表中结点 A 的后继结点 B，指针 s 指向被插入的结点 X，则在结点 A 和结点 B 插入结点 X 的操作序列为 ( B )。  
A、 $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s;$  B、 $q \rightarrow \text{next} = s; s \rightarrow \text{next} = p;$   
C、 $p \rightarrow \text{next} = s \rightarrow \text{next}; s \rightarrow \text{next} = p;$  D、 $p \rightarrow \text{next} = s; s \rightarrow \text{next} = q;$
4. 设输入序列为 1、2、3、4、5、6，则通过栈的作用后可以得到的输出序列为 ( B )。  
A、5, 3, 4, 6, 1, 2 B、3, 2, 5, 6, 4, 1  
C、3, 1, 2, 5, 4, 6 D、1, 5, 4, 6, 2, 3
5. 设有一个 10 阶的下三角矩阵 A (包括对角线)，按照从上到下、从左到右的顺序存储到连续的 55 个存储单元中，每个数组元素占 1 个字节的存储空间，则  $A[5][4]$  地址与  $A[0][0]$  的地址之差为 ( B )。  
A、10 B、19 C、28 D、55
6. 设一棵 m 叉树中有  $N_1$  个度数为 1 的结点， $N_2$  个度数为 2 的结点，……， $N_m$  个度数为 m 的结点，则该树中共有 ( D ) 个叶子结点。  
A、 $\sum_{i=1}^m (i-1)N_i$  B、 $\sum_{i=1}^m N_i$  C、 $\sum_{i=2}^m N_i$  D、 $1 + \sum_{i=2}^m (i-1)N_i$
7. 二叉排序树中左子树上所有结点的值均 ( A ) 根结点的值。  
A、< B、> C、= D、!=
8. 设一组权值集合  $W=(15, 3, 14, 2, 6, 9, 16, 17)$ ，要求根据这些权值集合构造一棵哈夫曼树，则这棵哈夫曼树的带权路径长度为 ( D )。  
A、129 B、219 C、189 D、229
9. 设有 n 个关键字具有相同的 Hash 函数值，则用线性探测法把这 n 个关键字映射到 HASH 表中需要做 ( D ) 次线性探测。  
A、 $n^2$  B、 $n(n+1)$  C、 $n(n+1)/2$  D、 $n(n-1)/2$
10. 设某棵二叉树中只有度数为 0 和度数为 2 的结点且度数为 0 的结点数为 n，则这棵二叉树中共有 ( C ) 个结点。

- A、  $2n$                       B、  $n+1$                       C、  $2n-1$                       D、  $2n+1$
11. 设一组初始记录关键字的长度为 8, 则最多经过 ( B ) 趟插入排序可以得到有序序列。  
A、 6                      B、 7                      C、 8                      D、 9
12. 设一组初始记录关键字序列为(Q, H, C, Y, P, A, M, S, R, D, F, X), 则按字母升序的第一趟冒泡排序结束后的结果是 ( D )。  
A、 F, H, C, D, P, A, M, Q, R, S, Y, X  
B、 P, A, C, S, Q, D, F, X, R, H, M, Y  
C、 A, D, C, R, F, Q, M, S, Y, P, H, X  
D、 H, C, Q, P, A, M, S, R, D, F, X, Y

## 二、填空题

- 设需要对 5 个不同的记录关键字进行排序, 则至少需要比较 (4) 次, 至多需要比较 (10) 次。
- 快速排序算法的平均时间复杂度为 (  $O(n\log_2 n)$  ), 直接插入排序算法的平均时间复杂度为 (  $O(n^2)$  )。
- 设二叉排序树的高度为  $h$ , 则在该树中查找关键字  $key$  最多需要比较 (  $n$  ) 次。
- 设在长度为 20 的有序表中进行二分查找, 则比较一次查找成功的结点数有 (1) 个, 比较两次查找成功有结点数有 (2) 个。
- 设一棵  $m$  叉树的结点个数为  $n$ , 用多重链表表示其存储结构, 则该树中有 (  $n(m-1)+1$  ) 个空指针域。
- 设指针变量  $p$  指向单链表中结点  $A$ , 则删除结点  $A$  的语句序列为:  
 $q=p \rightarrow next$ ;  $p \rightarrow data=q \rightarrow data$ ;  $p \rightarrow next=(q \rightarrow next)$ ;  $free(q)$ ;
- 数据结构从逻辑上划分为三种基本类型: (线性结构)、(树型结构) 和 (图型结构)。
- 设无向图  $G$  中有  $n$  个顶点  $e$  条边, 则用邻接矩阵作为图的存储结构进行深度优先或广度优先遍历时的时间复杂度为 (  $O(n^2)$  ); 用邻接表作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 (  $O(n+e)$  )。
- 设散列表的长度为 8, 散列函数  $H(k)=k \% 7$ , 用线性探测法解决冲突, 则根据一组初始关键字序列 (8, 15, 16, 22, 30, 32) 构造出的散列表的平均查找长度是 (8/3)。
- 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 则第 3 趟冒泡排序结束后的结果为 ( (38, 13, 27, 10, 65, 76, 97) )。
- 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 则第 3 趟简单选择排序后的结果为 (10, 13, 27, 76, 65, 97, 38)。
- 设有向图  $G$  中的有向边的集合  $E=\{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 4 \rangle, \langle 4, 5 \rangle, \langle 5, 3 \rangle, \langle 4, 6 \rangle, \langle 6, 5 \rangle \}$ , 则该图的一个拓扑序列为 (124653)。
- 下面程序段的功能是建立二叉树的算法, 请在下划线处填上正确的内容。  

```
typedef struct node {int data; struct node *lchild; (struct node *rchild ) ;} bitree;
void createbitree(bitree *&bt)
{
    scanf("%c", &ch);
    if(ch=='#') ( bt=0 ); else
        {bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch; (createbitree(bt->lchild));
          createbitree(bt->rchild);}
}
```
- 下面程序段的功能是利用从尾部插入的方法建立单链表的算法, 请在下划线处填上正确的内容。



```

typedef struct node {int data; struct node *next;} llist;
void llistcreate( (llist) *&head )
{
    for (i=1;i<=n;i++)
    {
        p=(llist *)malloc(sizeof(llist));scanf("%d",&(p->data));p->next=0;
        if(i==1)head=q=p;else {q->next=p; ( q=p ) ;}
    }
}

```

### 三、算法设计题

1. 设计在链式存储结构上合并排序的算法。

参考答案:

```

void mergelklist(llist *ha,llist *hb,llist *&hc)
{
    llist *s=hc=0;
    while(ha!=0 && hb!=0)
        if(ha->data<hb->data){if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}
        else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}
    if(ha==0) s->next=hb; else s->next=ha;
}

```

2. 设计在二叉排序树上查找结点 X 的算法。

参考答案:

```

bitree *bstsearch1(bitree *t, int key)
{
    bitree *p=t;
    while(p!=0) if (p->key==key) return(p);else if (p->key>key)p=p->lchild; else p=p->rchild;
    return(0);
}

```

3. 设关键字序列  $(k_1, k_2, \dots, k_{n-1})$  是堆, 设计算法将关键字序列  $(k_1, k_2, \dots, k_{n-1}, x)$  调整为堆。

参考答案:

```

void adjustheap(int r[],int n)
{
    int j=n,i=j/2,temp=r[j-1];
    while (i>=1) if (temp>=r[i-1])break; else {r[j-1]=r[i-1]; j=i; i=i/2;}
    r[j-1]=temp;
}

```