

《数据结构》期末考试试题及答案

(2003-2004 学年第 2 学期)

一、 单项选择题

1. 对于一个算法，当输入非法数据时，也要能作出相应的处理，这种要求称为（ C ）。

- (A)、正确性 (B)、可行性 (C)、健壮性 (D)、输入性

2. 设 S 为 C 语言的语句，计算机执行下面算法时，算法的时间复杂度为（ D ）。

```
for(i=n-1; i>=0; i--)
```

```
    for(j=0; j<i; j++) S;
```

- (A)、 $n^2$  (B)、 $O(n\lg n)$  (C)、 $O(n)$  (D)、 $O(n^2)$

3. 折半查找法适用于（ A ）。

- (A)、有序顺序表 (B)、有序单链表

- (C)、有序顺序表和有序单链表都可以 (D)、无限制

4. 顺序存储结构的优势是（ D ）。

- (A)、利于插入操作 (B)、利于删除操作

- (C)、利于顺序访问 (D)、利于随机访问

5. 深度为 k 的完全二叉树，其叶子结点必在第（ C ）层上。

- (A)、k-1 (B)、k (C)、k-1 和 k (D)、1 至 k

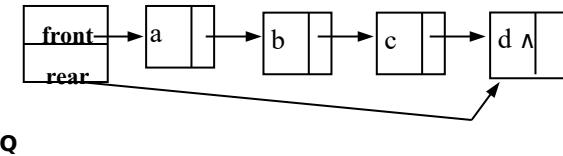
6. 具有 60 个结点的二叉树，其叶子结点有 12 个，则度数 1 的结点数为（ 1-37 2-11 ）

- (A)、11 (B)、13 (C)、48 (D)、37

7. 图的 Depth-First Search(DFS)遍历思想实际上是二叉树（ A ）遍历方法的推广。

- (A)、先序 (B)、中序 (C)、后序 (D)、层序

8. 在下列链队列 Q 中，元素 a 出队的操作序列为（ B ）



- (A)、p=Q.front->next; p->next= Q.front->next;

- (B)、p=Q.front->next; Q.front->next=p->next;

- (C)、p=Q.rear->next; p->next= Q.rear->next;

- (D)、p=Q->next; Q->next=p->next;

!!!9. Huffman 树的带权路径长度 WPL 等于（ AC ）

- (A)、除根结点之外的所有结点权值之和 (B)、所有结点权值之和

- (C)、各叶子结点的带权路径长度之和 (D)、根结点的值

10. 索引二叉链表是利用（ C ）域存储后继结点的地址。

- (A)、lchild (B)、data (C)、rchild (D)、root

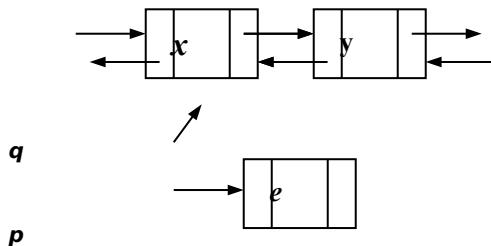
二、填空题

1. 逻辑结构决定了算法的 设计，而存储结构决定了算法的 实现。

2. !!! 栈和队列都是一种 顺序存取 的线性表，栈的插入和删除只能在 栈顶 进行。

3. 线性表  $(a_1, a_2, \dots, a_n)$  的顺序存储结构中，设每个单元的长度为  $L$ ，元素  $a_i$  的存储地址  $LOC(a_i)$  为  $LOC(a_0) + L*i$

4. 已知一双向链表如下(指针域名为 **next** 和 **prior**)：

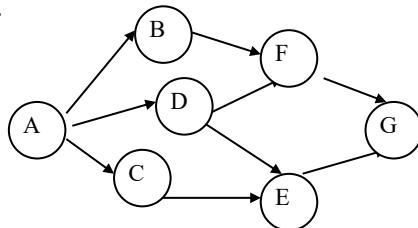


现将  $p$  所指的结点插入到  $x$  和  $y$  结点之间，其操作步骤为：  
 $p->next=y$ ；  
 $p->prior=x$ ；  
 $x->next=p$ ；  
 $y->prior=p$ ；

5.  $n$  个结点无向完全图的边数为  $n*(n-1)/2$ ，

$n$  个结点的生成树的边数为  $n-1$ 。

6. 已知一有向无环图如下：



任意写出二种拓扑排序序列： **ABCDFEG**、 **ABCDEFG**。

7. 已知二叉树的中序遍历序列为 **BCA**，后序遍历序列为 **CBA**，则该二叉树的先序遍历序列为 **ABC**，层序遍历序列为 **ABC**。

### 三、应用题

1. 设散列函数  $H(k)=k \% 13$ ，设关键字系列为  $\{22, 12, 24, 6, 45, 7, 8, 13, 21\}$ ，要求用线性探测法处理冲突。**(6 分)**

(1) 构造 HASH 表。

(2) 分别求查找成功和不成功时的平均查找长度。

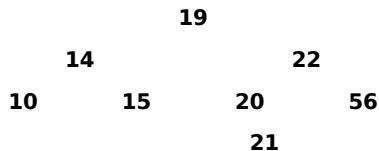
0	1	2	3	4	5	6	7	8	9	10	11	12
13	21					6	45	7	22	8	24	12

成功：  $(1+1+1+1+2+2+3+1+7)/9=19/9$

失败：  $(3+2+1+1+1+10+9+8+7+6+5+4)/13=58/13$

2. 给定表 **(19, 14, 22, 15, 20, 21, 56, 10)**。**(8 分)**

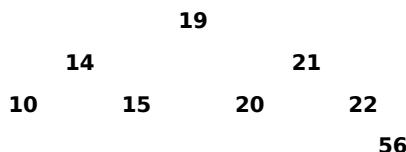
- (1) 按元素在表中的次序, 建立一棵二叉排序树



- (2) 对(1)中所建立的二叉排序树进行中序遍历, 写出遍历序列。

**10,14,15,19,20,21,22,56**

- (3) 画出对(2)中的遍历序列进行折半查找过程的判定树。



3. 已知二个稀疏矩阵 A 和 B 的压缩存储三元组表如下:

A		
i	j	v
1	3	-5
2	4	6
2	5	2
4	2	-1
5	2	9

i	j	v
2	5	2
3	3	7
4	1	3
5	2	-9
5	5	8

- 写出 A-B 压缩存储的三元组表。 (5 分)

i	j	v
1	3	-5
2	4	6
3	3	-7
4	1	-3
4	2	-1
5	2	18
5	5	-8

4. 已知一维数组中的数据为 (18,12,25,53,18), 试写出插入排序 (升序) 过程。并指出具有 n 个元素的插入排序的时间复杂度是多少? (5 分)

**18**

**12 18**

**12 18 25**

**12 18 25 53**

**12 18 18 25 53**

**O(n^2)**

**5.** ???已知一网络的邻接矩阵如下，求从顶点 **A** 开始的最小生成树。**(8 分，要有过程)**

**A B C D E F**

<b>A</b>	$\infty$	6	5	1	$\infty$	$\infty$
<b>B</b>	6	$\infty$	$\infty$	5	3	$\infty$
<b>C</b>	5	$\infty$	$\infty$	7	$\infty$	2
<b>D</b>	1	5	7	$\infty$	6	4
<b>E</b>	$\infty$	3	$\infty$	6	$\infty$	6
<b>F</b>	$\infty$	$\infty$	2	4	6	$\infty$

**(1)** 求从顶点 **A** 开始的最小生成树。

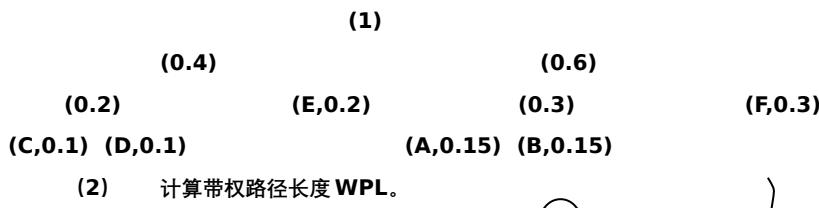
**(2)** 分别画出以 **A** 为起点的 **DFS** 生成树和 **BFS** 生成树。

**6.** 已知数据六个字母及在通信中出现频率如下表：

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>0.15</b>	<b>0.15</b>	<b>0.1</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>

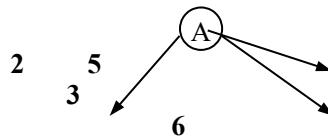
把这些字母和频率作为叶子结点及权值，完成如下工作**(7 分，要有过程)**。

**(1)** 画出对应的 **Huffman** 树。

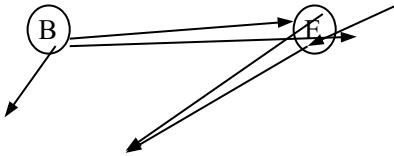


<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>100</b>	<b>101</b>	<b>000</b>	<b>001</b>	<b>01</b>	<b>11</b>

**7.** ???已知有如下的有向网：



4 10 6 1 2



求顶点 **A** 到其它各顶点的最短路径（采用 **Dijkstra** 算法，要有过程）。 (6 分)

三、设计题 (30 分，每题 10 分，用 C 语言写出算法，做在答题纸上)

1. 已知线性表  $(a_1, a_2, \dots, a_n)$  以顺序存储结构为存储结构，其类型定义如下：

```
#define LIST_INIT_SIZE 100 //顺序表初始分配容量
typedef struct {
    Elemtpe *elem; //顺序存储空间基址
    int length; //当前长度 (存储元素个数)
} SqList;
```

设计一个算法，删除其元素值为  $x$  的结点（假若  $x$  是唯一的）。并求出其算法的平均时间复杂度。其算法函数头部如下：

```
Status ListDelete(SqList &L, Elemtpe x)
{
    For(int i=0;i<length;i++) if(L.elem[i]==x){
        For(int j=i;j<length-1;j++){
            L.elem[j]=L.elem[j+1];
        }
        Length--;
        Return;
    }
}
```

2. 设顺序栈如左图所示。

其中结点定义如下：

top

```
typedef struct {
```

```
    Elemtpe *base; //栈底指针
```

```
    Elemtpe *top; //栈顶指针
```

```
} Stack;
```

设计算法，将栈顶元素出栈存入  $e$  中。

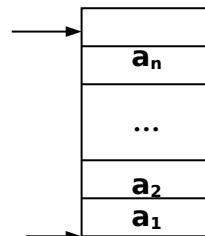
base

```
Void Pop(Stack& stk, Elemtpe& e){
```

```
    if(stk.top==stk.base) return;
```

```
    e=*&--stk.top;
```

```
}
```



3. 设二叉链树的类型定义如下：

```
typedef int Elemtpe;  
typedef struct node{  
    Elemtpe data;  
    struct node *lchild, *rchild;  
}BinNode, *BinTree;
```

试写出求该二叉树叶子结点数的算法：

```
Status CountLeaves(BinTree &root,int &n)  
{//n is the number of leaves
```

Return

}

答案：

选择题（每题1分）

1、C 2、D 3、A 4、D 5、C 6、D 7、A 8、B 9、C 10、C

一、填空题

1. 设计、实现

2. 特殊、栈顶

3. LOC (a1) +(i-1)\*L

4. p->next=q->next;q->next->prior=p; **q->next=p;p->prior=q;**

5. n(n-1)/2、n-1

6. ADCBFEG、ABCDEFFG

7. ABC、ABC

二、应用题

1 (1) Hash表 (4分)

地址	0	1	2	3	4	5	6	7	8	9	10	11	12
关键安	13	21					6	45	7	22	8	24	12
探测次数	1	7					1	2	3	1	3	1	1

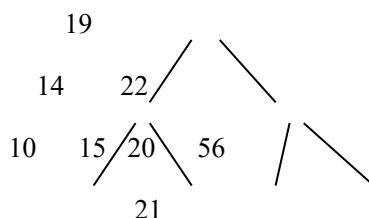
(2) 查找成功的平均查找长度：(1分)

$$(5*1+1*2+2*3+1*7) /9=20/9$$

查找不成功的平均查找长度：(1分)

$$(2+1+9+8+7+6+5+4+3+2+1) /13=$$

2 (1) 、构造 (3分)



(2)、10 14 15 19 20 21 22 56 (2 分)

(3) 、 (3 分)

3、(5 分, 每行 0.5)

i	j	v
1	3	-5
2	4	6
3	3	7
4	1	3
4	2	-1
5	2	18
5	5	8

4、初始关键字: [18] 12 25 53 18

第一 趟: [12 18] 25 53 18

第二 趟: [12 18 25] 53 18



第三 趟: [12 18 25 53] 18

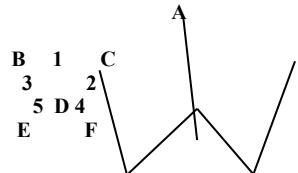
第四 趟: [12 18 18 25 53]

←(4分)

$O(n^2)$  (1 分)。

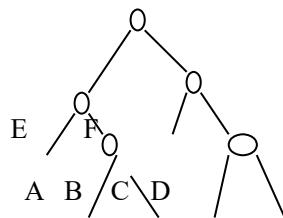
5、7 分

(1) 4 分



(2) 4 分

6、(1) 3 分



(2)  $WPL = 0.1*3 + 0.1*3 + 0.2*2 + 0.15*3 + 0.15*3 + 0.3*2 = \quad \text{(1 分)}$

(3) A: 010 B: 011 C: 110 D: 111 E: 00 F: 10 (3 分)

12、A-B: (A、B) 1 分

A-C: (A、D、C) 2 分

A-D: (A、D) 1 分

A-E: (A、D、E) 2 分

三, 设计题 (20 分)

1、(10 分)

```
Status ListDelete(Sqlist &L,ElemType x)
{
    int i,j;
    for(i=0;i<L->length;i++)
        if(L->elem[i]==x) break;
    if(i==L->length) return ERROR;
    for(j=i;j<L->length-1;j++)
        L->elem[j]=L->elem[j+1];
    L->length--;
}
```

(8 分)

平均时间复杂度: (2 分)

设元素个数记为 n, 则平均时间复杂度为:

$$E = \frac{1}{n} \sum_{i=1}^n (n-i) = \frac{n-1}{2}$$

2 (10 分)

```
void pop(Stack &S,Elemtype &e)
```

```
{
```

```
if(S.top==S.base) return ERROR;
```

```
S.top--;
```

```
e=*S.top;
```

```
}
```

2、 (10 分)

```
voidCountLeaves(BinTree T,int &n)
```

```
{
```

```
if(T)
```

```
{
```

```
if((!T->lchild)&&(!T->rchild)) n++;
```

```
CountLeaves (T->lchild,n);
```

```
CountLeaves (T->rchild,n);
```

```
}
```

```
}
```

## 习题 1

### 一、单项选择题

1. 数据结构是指 ( A ) 。

A.数据元素的组织形式 B.数据类型

C.数据存储结构 D.数据定义

???2. 数据在计算机存储器内表示时，物理地址与逻辑地址不相同的，称之为 ( C ) 。

A.存储结构 B.逻辑结构

C.链式存储结构 D.顺序存储结构

3. 树形结构是数据元素之间存在一种 ( D ) 。

A.一对一关系 B.多对多关系

C.多对一关系 D.一对多关系

4. 设语句 `x++` 的时间是单位时间，则以下语句的时间复杂度为 (  $n^2$  ) 。

```
for(i=1; i<=n; i++)
```

```
for(j=i; j<=n; j++)
```

```
x++;
```

A.O(1) B.O( ) C.O(n) D.O( )

???5. 算法分析的目的是 (A) ， 算法分析的两个主要方面是 (AD) 。

- (1) A.找出数据结构的合理性      B.研究算法中的输入和输出关系  
C.分析算法的效率以求改进      D.分析算法的易懂性和文档性
- (2) A.空间复杂度和时间复杂度      B.正确性和简明性  
C.可读性和文档性      D.数据复杂性和程序复杂性
6. 计算机算法指的是 (C) , 它具备输入, 输出和 (B) 等五个特性。
- (1) A.计算方法      B.排序方法  
C.解决问题的有限运算序列      D.调度方法
- (2) A.可行性, 可移植性和可扩充性      B.可行性, 确定性和有穷性  
C.确定性, 有穷性和稳定性      D.易读性, 稳定性和安全性
7. 数据在计算机内有链式和顺序两种存储方式, 在存储空间使用的灵活性上, 链式存储比顺序存储要 (B) 。
- A.低    B.高    C.相同    D.不好说

???8. 数据结构作为一门独立的课程出现是在 ( ) 年。

A.1946    B.1953    C.1964    D.1968

9. 数据结构只是研究数据的逻辑结构和物理结构, 这种观点 (B) 。

A.正确    B.错误

C.前半句对, 后半句错    D.前半句错, 后半句对

10. 计算机内部数据处理的基本单位是 (B) 。

A.数据    B.数据元素    C.数据项    D.数据库

## 二、填空题

1. 数据结构按逻辑结构可分为两大类, 分别是\_\_\_\_\_ ? 和\_\_\_\_\_。
2. 数据的逻辑结构有四种基本形态, 分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
3. 线性结构反映结点间的逻辑关系是\_\_\_\_\_线性\_\_\_\_\_的, 非线性结构反映结点间的逻辑关系是\_\_\_\_\_非线性\_\_\_\_\_的。
4. 一个算法的效率可分为\_\_\_\_\_时间\_\_\_\_\_效率和\_\_\_\_\_空间\_\_\_\_\_效率。
5. 在树型结构中, 树根结点没有\_\_\_\_\_父\_\_\_\_\_结点, 其余每个结点的有且只有\_\_\_\_\_1\_\_\_\_\_个前趋驱动结点; 叶子结点没有\_\_\_\_\_子\_\_\_\_\_结点; 其余每个结点的后续结点可以\_\_\_\_\_。
6. 在图型结构中, 每个结点的前趋结点数和后续结点数可以\_\_\_\_\_。
7. 线性结构中元素之间存在\_\_\_\_\_线性\_\_\_\_\_关系; 树型结构中元素之间存在\_\_\_\_非线性\_\_\_\_关系; 图型结构中元素之间存在\_\_\_\_\_非线性\_\_\_\_\_关系。
8. 下面程序段的时间复杂度是\_\_\_\_\_  $n^2$  \_\_\_\_\_。  

```
for(i=0;i<n;i++)  
for(j=0;j<n;j++)  
A[i][j]=0;
```
9. 下面程序段的时间复杂度是\_\_\_\_\_  $\sqrt{n}$  \_\_\_\_\_。  

```
i=s=0;
```

```
while(s<n)
{ i++;
  s+=i;
}
```

10. 下面程序段的时间复杂度是\_\_\_\_\_  $n^2$  \_\_\_\_\_。

```
s=0;
for(i=0;i<n;i++)
for(j=0;j<n;j++)
s+=B[i][j];
sum=s;
```

11. 下面程序段的时间复杂度是\_\_\_\_\_  $\log_3 n$  \_\_\_\_\_。

```
i=1;
while(i<=n)
i=i*3;
```

12. 衡量算法正确性的标准通常是\_\_\_\_\_。

13. 算法时间复杂度的分析通常有两种方法，即\_\_\_\_模拟\_\_\_\_和\_\_\_\_理论推导\_\_\_\_的方法，通常我们对算法求时间复杂度时，采用后一种方法。

三、求下列程序段的时间复杂度。

```
1. x=0;
for(i=1;i<n;i++)
for(j=i+1;j<=n;j++)
x++;
```

$n^2$

```
2. x=0;
for(i=1;i<n;i++)
for(j=1;j<=n-i;j++)
x++;
```

$n^2$

```
3. int i,j,k;
for(i=0;i<n;i++)
for(j=0;j<=n;j++)
{ c[i][j]=0;
for(k=0;k<n;k++)
  c[i][j]=a[i][k]*b[k][j]
}
```

$n^3$

```
4. i=n-1;  
while((i>=0)&&A[i]!=k))  
j-;  
return (i);  
  
n  
5. fact(n)  
{ if(n<=1)  
return (1);  
else  
return (n*fact(n-1));  
}  
  
n
```

## 习题 1 参考答案

### 一、单项选择题

1. A 2. C 3. D 4. B 5. C、A 6. C、B 7. B 8. D 9. B 10. B

### 二、填空题

1. 线性结构, 非线性结构
2. 集合, 线性, 树, 图
3. 一对一, 一对多或多对多
4. 时间, 空间
5. 前趋, 一, 后继, 多
6. 有多个
7. 一对一, 一对多, 多对多
8. O( )
9. O( )
10. O( )
11. O(log n)

12. 程序对于精心设计的典型合法数据输入能得出符合要求的结果。

13. 事后统计, 事前估计

### 三、算法设计题

1. O( ) 2. O( ) 3. O(n) 4. O(n) 5. O(n)

## 习题 2

### 一、单项选择题

1. 线性表是\_\_\_\_A\_\_\_\_。

A . 一个有限序列, 可以为空 B . 一个有限序列, 不可以为空

- C . 一个无限序列，可以为空 D . 一个无限序列，不可以为空
2. 在一个长度为 n 的顺序表中删除第 i 个元素( $0 \leq i \leq n$ )时，需向前移动    个元素。  
A . n-i B . n-i+1 C . n-i-1 D . i
3. 线性表采用链式存储时，其地址D\_\_\_\_\_。  
A . 必须是连续的 B . 一定是不连续的  
C . 部分地址必须是连续的 D . 连续与否均可以
4. 从一个具有 n 个结点的单链表中查找其值等于 x 的结点时，在查找成功的情况下，需平均比较 C\_\_\_\_\_个元素结点。  
A .  $n/2$  B . n C .  $(n+1)/2$  D .  $(n-1)/2$
5. 在双向循环链表中，在 p 所指的结点之后插入 s 指针所指的结点，其操作是D\_\_\_\_\_。  
A . p->next=s; s->prior=p;  
p->next->prior=s; s->next=p->next;  
B . s->prior=p; s->next=p->next;  
p->next=s; p->next->prior=s;  
C . p->next=s; p->next->prior=s;  
s->prior=p; s->next=p->next;  
D . s->prior=p; s->next=p->next;  
p->next->prior=s; p->next=s;
6. 设单链表中指针 p 指向结点 m，若要删除 m 之后的结点（若存在），则需修改指针的操作为 A\_\_\_\_\_。  
A . p->next=p->next->next; B . p=p->next;  
C . p=p->next->next; D . p->next=p;
7. 在一个长度为 n 的顺序表中向第 i 个元素( $0 < i < n+1$ )之前插入一个新元素时，需向后移动B\_\_\_\_\_个元素。  
A . n-i B . n-i+1 C . n-i-1 D . i
8. 在一个单链表中，已知 q 结点是 p 结点的前趋结点，若在 q 和 p 之间插入 s 结点，则须执行 B  
A . s->next=p->next; p->next=s  
B . q->next=s; s->next=p  
C . p->next=s->next; s->next=p  
D . p->next=s; s->next=q
9. 以下关于线性表的说法不正确的是D\_\_\_\_\_。  
A . 线性表中的数据元素可以是数字、字符、记录等不同类型。  
B . 线性表中包含的数据元素个数不是任意的。  
C . 线性表中的每个结点都有且只有一个直接前趋和直接后继。  
D . 存在这样的线性表：表中各结点都没有直接前趋和直接后继。
10. 线性表的顺序存储结构是一种C\_\_\_\_\_的存储结构。  
A . 随机存取 B . 顺序存取 C . 索引存取 D . 散列存取
11. 在顺序表中，只要知道D\_\_\_\_\_，就可在相同时间内求出任一结点的存储地址。

- A . 基地址    B . 结点大小  
C . 向量大小    D . 基地址和结点大小
12. 在等概率情况下，顺序表的插入操作要移动\_\_\_\_B\_\_\_\_结点。  
A . 全部    B . 一半  
C . 三分之一    D . 四分之一
13. 在\_\_\_\_C\_\_\_\_运算中，使用顺序表比链表好。  
A . 插入    B . 删除  
C . 根据序号查找    D . 根据元素值查找
14. 在一个具有 n 个结点的有序单链表中插入一个新结点并保持该表有序的时间复杂度是\_\_\_\_B\_\_\_\_。  
A . O(1)    B . O(n)  
C . O(n<sup>2</sup>)    D . O(log2n)
15. 设有一个栈，元素的进栈次序为 A, B, C, D, E, 下列是不可能的出栈序列\_\_\_\_C\_\_\_\_。  
A . A, B, C, D, E    B . B, C, D, E, A  
C . E, A, B, C, D    D . E, D, C, B, A
16. 在一个具有 n 个单元的顺序栈中，假定以地址低端（即 0 单元）作为栈底，以 top 作为栈顶指针，当做出栈处理时，top 变化为\_\_\_\_C\_\_\_\_。  
A . top 不变    B . top=0    C . top--    D . top++
17. 向一个栈顶指针为 hs 的链栈中插入一个 s 结点时，应执行\_\_\_\_C\_\_\_\_。  
A . hs->next=s;  
B . s->next=hs; hs=s;  
C . s->next=hs->next;hs->next=s;  
D . s->next=hs; hs=hs->next;
18. 在具有 n 个单元的顺序存储的循环队列中，假定 front 和 rear 分别为队头指针和队尾指针，则判断队满的条件为\_\_\_\_D\_\_\_\_。  
A . rear%n= = front    B . (front+l) %n= = rear  
C . rear%n -1= = front    D . (rear+l)%n= = front
19. 在具有 n 个单元的顺序存储的循环队列中，假定 front 和 rear 分别为队头指针和队尾指针，则判断队空的条件为\_\_\_\_B\_\_\_\_。  
A . rear%n= = front    B . (front+l)%n= rear  
C . rear= = front    D . (rear+l)%n= front
20. 在一个链队列中，假定 front 和 rear 分别为队首和队尾指针，则删除一个结点的操作为\_\_\_\_A\_\_\_\_。  
A . front=front->next    B . rear=rear->next  
C . rear=front->next    D . front=rear->next
- 二、填空题
- 线性表是一种典型的\_\_\_\_线性数据\_\_\_\_结构。
  - 在一个长度为 n 的顺序表的第 i 个元素之前插入一个元素，需要后移\_\_\_\_n-i+1\_\_\_\_个元素。
  - 顺序表中逻辑上相邻的元素的物理位置\_\_\_\_相邻\_\_\_\_。
  - 要从一个顺序表删除一个元素时，被删除元素之后的所有元素均需\_\_\_\_前移\_\_\_\_一个位置，移动过程

- 是从\_\_\_\_删除位置\_\_\_\_向\_\_\_\_尾部\_\_\_\_依次移动每一个元素。
5. 在线性表的顺序存储中，元素之间的逻辑关系是通过\_\_\_\_\_决定的；在线性表的链接存储中，元素之间的逻辑关系是通过\_\_\_\_\_决定的。
6. 在双向链表中，每个结点含有两个指针域，一个指向\_\_\_\_\_结点，另一个指向\_\_\_\_\_结点。
7. 当对一个线性表经常进行存取操作，而很少进行插入和删除操作时，则采用\_\_\_\_\_存储结构为宜。相反，当经常进行的是插入和删除操作时，则采用\_\_\_\_\_存储结构为宜。
8. 顺序表中逻辑上相邻的元素，物理位置\_\_\_\_\_相邻，单链表中逻辑上相邻的元素，物理位置\_\_\_\_\_相邻。
9. 线性表、栈和队列都是\_\_\_\_\_结构，可以在线性表的\_\_\_\_\_位置插入和删除元素；对于栈只能在\_\_\_\_\_位置插入和删除元素；对于队列只能在\_\_\_\_\_位置插入元素和在\_\_\_\_\_位置删除元素。
10. 根据线性表的链式存储结构中每个结点所含指针的个数，链表可分为\_\_\_\_\_和\_\_\_\_\_；而根据指针的联接方式，链表又可分为\_\_\_\_\_和\_\_\_\_\_。
11. 在单链表中设置头结点的作用是\_\_\_\_\_。
12. 对于一个具有 n 个结点的单链表，在已知的结点 p 后插入一个新结点的时间复杂度为\_\_\_\_\_，在给定值为 x 的结点后插入一个新结点的时间复杂度为\_\_\_\_\_。
13. 对于一个栈作进栈运算时，应先判别栈是否为\_\_\_\_\_，作退栈运算时，应先判别栈是否为\_\_\_\_\_。当栈中元素为 m 时，作进栈运算时发生上溢，则说明栈的可用最大容量为\_\_\_\_\_。为了增加内存空间的利用率和减少发生上溢的可能性，由两个栈共享一片连续的内存空间时，应将两栈的\_\_\_\_\_分别设在这片内存空间的两端，这样只有当\_\_\_\_\_时才产生上溢。
14. 设有一空栈，现有输入序列 1, 2, 3, 4, 5，经过 push, push, pop, push, pop, push, push 后，输出序列是\_\_\_\_23\_\_\_\_\_。
15. 无论对于顺序存储还是链式存储的栈和队列来说，进行插入或删除运算的时间复杂度均相同为\_\_\_\_\_。

### 三、简答题

1. 描述以下三个概念的区别：头指针，头结点，表头结点。
2. 线性表的两种存储结构各有哪些优缺点？
3. 对于线性表的两种存储结构，如果有 n 个线性表同时并存，而且在处理过程中各表的长度会动态发生变化，线性表的总数也会自动改变，在此情况下，应选用哪一种存储结构？为什么？
4. 对于线性表的两种存储结构，若线性表的总数基本稳定，且很少进行插入和删除操作，但要求以最快的速度存取线性表中的元素，应选用何种存储结构？试说明理由。
5. 在单循环链表中设置尾指针比设置头指针好吗？为什么？
6. 假定有四个元素 A, B, C, D 依次进栈，进栈过程中允许出栈，试写出所有可能的出栈序列。
7. 什么是队列的上溢现象？一般有几种解决方法，试简述之。
8. 下述算法的功能是什么？

```
LinkList *Demo(LinkList *L)
{
    // L 是无头结点的单链表
    LinkList *q, *p;
    if(L && L->next)
```

```

{ q=L; L=L->next; p=L;
    while (p->next)
p=p->next;
    p->next=q; q->next=NULL;
}
return (L);
}

```

#### 四、算法设计题

1. 设计在无头结点的单链表中删除第  $i$  个结点的算法。
2. 在单链表上实现线性表的求表长  $ListLength(L)$  运算。
3. 设计将带表头的链表逆置算法。
4. 假设有一个带表头结点的链表，表头指针为  $head$ ，每个结点含三个域：  $data$ ,  $next$  和  $prior$ 。其中  $data$  为整型数域，  $next$  和  $prior$  均为指针域。现在所有结点已经由  $next$  域连接起来，试编一个算法，利用  $prior$  域（此域初值为  $NULL$ ）把所有结点按照其值从小到大的顺序链接起来。
5. 已知线性表的元素按递增顺序排列，并以带头结点的单链表作为存储结构。试编写一个删除表中所有值大于  $min$  且小于  $max$  的元素（若表中存在这样的元素）的算法。
6. 已知线性表的元素是无序的，且以带头结点的单链表作为存储结构。设计一个删除表中所有值小于  $max$  但大于  $min$  的元素的算法。
7. 假定用一个单循环链表来表示队列（也称为循环队列），该队列只设一个队尾指针，不设队首指针，试编写下列各种运算的算法：
  - (1) 向循环链队列插入一个元素值为  $x$  的结点；
  - (2) 从循环链队列中删除一个结点。
8. 设顺序表  $L$  是一个递减有序表，试写一算法，将  $x$  插入其后仍保持  $L$  的有序性。

#### 习题 2 参考答案

##### 一、单项选择题

- 1 . A 2 . A 3 . D 4 . C 5 . D 6 . A 7 . B 8 . B 9 . C 10 . A 11 . D 12 . B 13 . C 14 . B  
 15 . C 16 . C 17 . B 18 . D 19 . C 20 . A

##### 二、填空题

- 1 . 线性
- 2 .  $n-i+1$
- 3 . 相邻
- 4 . 前移, 前, 后
- 5 . 物理存储位置, 链域的指针值
- 6 . 前趋, 后继
- 7 . 顺序, 链接
- 8 . 一定, 不一定
- 9 . 线性, 任何, 栈顶, 队尾, 队头
- 10 . 单链表, 双链表, 非循环链表, 循环链表

11. 使空表和非空表统一；算法处理一致

12.  $O(1)$ ,  $O(n)$

13. 栈满, 栈空,  $m$ , 栈底, 两个栈的栈顶在栈空间的某一位置相遇

14. 2、3

15.  $O(1)$

### 三、简答题

1. 头指针是指向链表中第一个结点（即表头结点）的指针；在表头结点之前附设的结点称为头结点；表头结点为链表中存储线性表中第一个数据元素的结点。若链表中附设头结点，则不管线性表是否为空表，头指针均不为空，否则表示空表的链表的头指针为空。

2. 线性表具有两种存储结构即顺序存储结构和链接存储结构。线性表的顺序存储结构可以直接存取数据元素，方便灵活、效率高，但插入、删除操作时将会引起元素的大量移动，因而降低效率；而在链接存储结构中内存采用动态分配，利用率高，但需增设指示结点之间关系的指针域，存取数据元素不如顺序存储方便，但结点的插入、删除操作较简单。

3. 应选用链接存储结构，因为链式存储结构是用一组任意的存储单元依次存储线性表中的各元素，这里存储单元可以是连续的，也可以是不连续的：这种存储结构对于元素的删除或插入运算是不需要移动元素的，只需修改指针即可，所以很容易实现表的容量的扩充。

4. 应选用顺序存储结构，因为每个数据元素的存储位置和线性表的起始位置相差一个和数据元素在线性表中的序号成正比的常数。因此，只要确定了其起始位置，线性表中的任一个数据元素都可随机存取，因此，线性表的顺序存储结构是一种随机存取的存储结构，而链表则是一种顺序存取的存储结构。

5. 设尾指针比设头指针好。尾指针是指向终端结点的指针，用它来表示单循环链表可以使得查找链表的开始结点和终端结点都很方便，设一带头结点的单循环链表，其尾指针为  $rear$ ，则开始结点和终端结点的位置分别是  $rear->next->next$  和  $rear$ ，查找时间都是  $O(1)$ 。若用头指针来表示该链表，则查找终端结点的时间为  $O(n)$ 。

6. 共有 14 种可能的出栈序列，即为：

ABCD, ABDC, ACBD, ACDB, BACD, ADCB, BADC, BCAD, BCDA, BDCA, CBAD,  
CBDA, CDBA, DCBA

7. 在队列的顺序存储结构中，设队头指针为  $front$ ，队尾指针为  $rear$ ，队列的容量（即存储的空间大小）为  $maxnum$ 。当有元素要加入队列（即入队）时，若  $rear=maxnum$ ，则会发生队列的上溢现象，此时就不能将该元素加入队列。对于队列，还有一种“假溢出”现象，队列中尚余有足够的空间，但元素却不能入队，一般是由于队列的存储结构或操作方式的选择不当所致，可以用循环队列解决。

一般地，要解决队列的上溢现象可有以下几种方法：

(1) 可建立一个足够大的存储空间以避免溢出，但这样做往往会造成空间使用率低，浪费存储空间。

(2) 要避免出现“假溢出”现象可用以下方法解决：

第一种：采用移动元素的方法。每当有一个新元素入队，就将队列中已有的元素向队头移动一个位置，假定空余空间足够。

第二种：每当删去一个队头元素，则可依次移动队列中的元素总是使  $front$  指针指向队列中的第一个位置。

第三种：采用循环队列方式。将队头、队尾看作是一个首尾相接的循环队列，即用循环数组实现，此时

队首仍在队尾之前，作插入和删除运算时仍遵循“先进先出”的原则。

8 . 该算法的功能是：将开始结点摘下链接到终端结点之后成为新的终端结点，而原来的第二个结点成为新的开始结点，返回新链表的头指针。

#### 四、算法设计题

1 . 算法思想为：

(1) 应判断删除位置的合法性，当  $i < 0$  或  $i > n - 1$  时，不允许进行删除操作；

(2) 当  $i = 0$  时，删除第一个结点；

(3) 当  $0 < i < n$  时，允许进行删除操作，但在查找被删除结点时，须用指针记住该结点的前趋结点。算法描述如下：

```
delete(LinkList *q,int i)
{ //在无头结点的单链表中删除第 i 个结点
    LinkList *p,*s;
    int j;
    if(i<0)
        printf("Can't delete");
    else if(i==0)
    {
        s=q;
        q=q->next;
        free(s);
    }
    else
    {
        j=0; s=q;
        while((j<i) && (s!=NULL))
        {
            p=s;
            s=s->next;
        }
        if (s==NULL)
            printf("Can't delete");
        else
        {
            p->next=s->next;
            free(s);
        }
    }
}
```

2 . 由于在单链表中只给出一个头指针，所以只能用遍历的方法来数单链表中的结点个数了。算法描述如下：

```
int ListLength ( LinkList *L )
```

```

{ //求带头结点的单链表的表长
    int len=0;
    ListList *p;
    p=L;
    while ( p->next!=NULL )
    { p=p->next;
        len++;
    }
    return (len);
}

```

3 . 设单循环链表的头指针为 head, 类型为 LinkList。逆置时需将每一个结点的指针域作以修改, 使其原前趋结点成为后继。如要更改 q 结点的指针域时, 设 s 指向其原前趋结点, p 指向其原后继结点, 则只需进行  $q->next=s$ ; 操作即可, 算法描述如下:

```

void invert(LinkList *head)
{ //逆置 head 指针所指向的单循环链表
linklist *p, *q, *s;
q=head;
p=head->next;
while (p!=head) //当表不为空时, 逐个结点逆置
{ s=q;
    q=p;
    p=p->next;
    q->next=s;
}
p->next=q;
}

```

4 . 定义类型 LinkList 如下:

```

typedef struct node
{
    int data;
    struct node *next,*prior;
}LinkList;

```

此题可采用插入排序的方法, 设 p 指向待插入的结点, 用 q 搜索已由 prior 域链接的有序表找到合适位置将 p 结点链入。算法描述如下:

```

insert (LinkList *head)
{
    LinkList *p,*s,*q;
    p=head->next; //p 指向待插入的结点, 初始时指向第一个结点
    while(p!=NULL)
    { s=head; // s 指向 q 结点的前趋结点

```

```

q=head->prior; //q 指向由 prior 域构成的链表中待比较的结点
while((q!=NULL) && (p->data>q->data)) //查找插入结点 p 的合适的插入位置
{
    s=q;
    q=q->prior;
}
s->prior=p;
p->prior=q; //结点 p 插入到结点 s 和结点 q 之间
p=p->next;
}
}

```

5 . 算法描述如下:

```

delete(LinkList *head, int max, int min)
{
    LinkList *p, *q;
    if (head==NULL)
    {
        q=head;
        p=head->next;
        while((p!=NULL) && (p->data<=min))
        {
            q=p;
            p=p->next;
        }
        while((p!=NULL) && (p->data<max))
            p=p->next;
        q->next=p;
    }
}

```

6 . 算法描述如下:

```

delete(LinkList *head, int max, int min)
{
    LinkList *p,*q;
    q=head;
    p=head->next;
    while (p!=NULL)
        if((p->data<=min) || (p->data>=max))
        {
            q=p;
            p=p->next;
        }
    else
        {
            q->next=p->next;
            free(p);
        }
}

```

```
p=q->next;
```

```
}
```

```
}
```

7. 本题是对一个循环链队列做插入和删除运算，假设不需要保留被删结点的值和不需要回收结点，算法描述如下：

(1) 插入（即入队）算法：

```
insert(LinkList *rear, elemtype x)
{
    //设循环链队列的队尾指针为 rear,x 为待插入的元素
    LinkList *p;
    p=(LinkList *)malloc(sizeof(LinkList));
    if(rear==NULL) //如为空队，建立循环链队列的第一个结点
    {
        rear=p;
        rear->next=p; //链接成循环链表
    }
    else //否则在队尾插入 p 结点
    {
        p->next=rear->next;
        rear->next=p;
        rear=p;
    }
}
```

(2) 删除（即出队）算法：

```
delete(LinkList *rear)
{
    //设循环链队列的队尾指针为 rear
    if (rear==NULL) //空队
        printf("underflow\n");
    if(rear->next==rear) //队中只有一个结点
        rear=NULL;
    else
        rear->next=rear->next->next; //rear->next 指向的结点为循环链队列的队头结点
}
```

8. 只要从终端结点开始往前找到第一个比 x 大(或相等)的结点数据，在这个位置插入就可以了。算法描述如下：

```
int InsertDecreaseList( SqList *L, elemtype x )
{
    int i;
    if ( (*L).len>= maxlen)
    {
        printf("overflow");
        return(0);
    }
```

```

for ( i=(*L).len ; i>0 && (*L).elem[ i-1 ] < x ; i-- )
    (*L).elem[ i ]=(*L).elem[ i-1 ] ; // 比较并移动元素
    (*L).elem[ i ] =x;
    (*L).len++;
return(1);
}

```

### 习题 3

#### 一、单项选择题

1. 空串与空格字符组成的串的区别在于（ ）。
  - A.没有区别
  - B.两串的长度不相等
  - C.两串的长度相等
  - D.两串包含的字符不相同
2. 一个子串在包含它的主串中的位置是指（ ）。
  - A.子串的最后那个字符在主串中的位置
  - B.子串的最后那个字符在主串中首次出现的位置
  - C.子串的第一个字符在主串中的位置
  - D.子串的第一个字符在主串中首次出现的位置
3. 下面的说法中，只有（ ）是正确的。
  - A.字符串的长度是指串中包含的字母的个数
  - B.字符串的长度是指串中包含的不同字符的个数
  - C.若 T 包含在 S 中，则 T 一定是 S 的一个子串
  - D.一个字符串不能说是其自身的一个子串
4. 两个字符串相等的条件是（ ）。
  - A.两串的长度相等
  - B.两串包含的字符相同
  - C.两串的长度相等，并且两串包含的字符相同
  - D.两串的长度相等，并且对应位置上的字符相同
5. 若 SUBSTR (S, i, k) 表示求 S 中从第 i 个字符开始的连续 k 个字符组成的子串的操作，则对于 S=“Beijing&Nanjing”，SUBSTR (S, 4, 5) =（ ）。
  - A. “ijing”
  - B. “jing&”
  - C. “ingNa”
  - D. “ing & N”
6. 若 INDEX (S, T) 表示求 T 在 S 中的位置的操作，则对于 S=“Beijing & Nanjing”，T=“jing”，INDEX (S, T) =（ ）。
  - A.2
  - B.3
  - C.4
  - D.5
7. 若 REPLACE (S, S1, S2) 表示用字符串 S2 替换字符串 S 中的子串 S1 的操作，则对于 S=“Beijing & Nanjing”，S1=“Beijing”，S2=“Shanghai”，REPLACE (S, S1, S2) =（ ）。
  - A. “Nanjing & Shanghai”
  - B. “Nanjing & Nanjing”

- C. “ShanghaiNanjing”      D. “Shanghai & Nanjing”
8. 在长度为 n 的字符串 S 的第 i 个位置插入另外一个字符串， i 的合法值应该是（ ）。  
A.  $i > 0$       B.  $i \leq n$   
C.  $1 \leq i \leq n$       D.  $1 \leq i \leq n+1$
9. 字符串采用结点大小为 1 的链表作为其存储结构，是指（ ）。  
A. 链表的长度为 1  
B. 链表中只存放 1 个字符  
C. 链表的每个链结点的数据域中不仅只存放了一个字符  
D. 链表的每个链结点的数据域中只存放了一个字符
- 二、填空题
1. 计算机软件系统中，有两种处理字符串长度的方法：一种是\_\_\_\_\_，第二种是\_\_\_\_\_。
  2. 两个字符串相等的充要条件是\_\_\_\_\_和\_\_\_\_\_。
  3. 设字符串 S1=“ABCDEF”， S2=“PQRS”，则运算  
 $S = \text{CONCAT} (\text{SUB} (S1, 2, \text{LEN} (S2)), \text{SUB} (S1, \text{LEN} (S2), 2))$  后的串值为\_\_\_\_\_。
  4. 串是指\_\_\_\_\_。
  5. 空串是指\_\_\_\_\_，空格串是指\_\_\_\_\_。

三、算法设计题

1. 设有一个长度为 s 的字符串，其字符顺序存放在一个一维数组的第 1 至第 s 个单元中（每个单元存放一个字符）。现要求从此串的第 m 个字符以后删除长度为 t 的子串， $m < s$ ， $t < (s-m)$ ，并将删除后的结果复制在该数组的第 s 单元以后的单元中，试设计此删除算法。
2. 设 s 和 t 是表示成单链表的两个串，试编写一个找出 s 中第 1 个不在 t 中出现的字符（假定每个结点只存放 1 个字符）的算法。

习题 3 参考答案

一、单项选择题

1. B 2. D 3. C 4. D 5. B 6. C 7. D 8. C 9. D

二、填空题

1. 固定长度，设置长度指针
2. 两个串的长度相等，对应位置的字符相等
3. “BCDEDE”
4. 含 n 个字符的有限序列 ( $n \geq 0$ )
5. 不含任何字符的串，仅含空格字符的字符串

三、算法设计题

1. 算法描述为：

```
int delete(r,s,t,m) //从串的第 m 个字符以后删除长度为 t 的子串
char r[ ];
int s,t,m;
```

```

{ int i,j;
  for(i=1;i<=m;i++)
    r[s+i]=r[i];
  for(j=m+t-i;j<=s;j++)
    r[s-t+j]=r[j];
  return (1);
} //delete

```

2. 算法思想为：

- (1) 链表 s 中取出一个字符；将该字符与单链表 t 中的字符依次比较；
- (2) 当 t 中有与从 s 中取出的这个字符相等的字符，则从 t 中取下一个字符重复以上比较；
- (3) 当 t 中没有与从 s 中取出的这个字符相等的字符，则算法结束。

设单链表类型为 LinkList；注意，此时类型 LinkList 中的 data 成分为字符类型。

```

LinkString find(s,t)
LinkString *s, *t;
{ LinkString *ps, *pt;
  ps=s;
  while(ps!=NULL)
  { pt=t;
    while((pt!=NULL)&&(ps->data!=pt->data))
      pt=pt->next;
    if(pt==NULL)
      ps=NULL;
    else
      { ps=ps->next;
        s=ps;
      }
  }
  return s;
} //find

```

#### 习题 4

##### 一、单项选择题

1. 设二维数组 A[0...m-1][0...n-1]按行优先顺序存储在内存中，第一个元素的地址为 p，每个元素占 k 个字节，则元素  $a_{ij}$  的地址为（ ）。  
 A.p + [i\*n+j-1]\*k    B.p + [(i-1)\*n+j-1]\*k  
 C.p + [(j-1)\*n+i-1]\*k    D.p + [j\*n+i-1]\*k
2. 已知二维数组 A $10 \times 10$  中，元素  $a_{20}$  的地址为 560，每个元素占 4 个字节，则元素  $a_{10}$  的地址为（ ）。

)。

A.520 B.522 C.524 D.518

3. 若数组 A[0...m][0...n]按列优先顺序存储，则  $a_{ij}$  地址为 ( )。

- A.LOC(a00)+[j\*m+i]      B. LOC(a00)+[j\*n+i]  
C.LOC(a00)+[(j-1)\*n+i-1]    D. LOC(a00)+[(j-1)\*m+i-1]

4. 若下三角矩阵  $An \times n$ , 按列顺序压缩存储在数组 Sa[0...(n+1)n/2]中, 则非零元素  $a_{ij}$  的地址为 ( )。

(设每个元素占 d 个字节)

- A. [(j-1)\*n- +i-1]\*d  
B. [(j-1)\*n- +i]\*d  
C. [(j-1)\*n- +i+1]\*d  
D. [(j-1)\*n- +i-2]\*d

5. 设有广义表  $D=(a,b,D)$ , 其长度为 ( ), 深度为 ( )。

- A.无穷大    B.3    C.2    D.5

6. 广义表  $A=(a)$ , 则表尾为 ( )。

- A.a    B.(( ))    C.空表    D.(a)

7. 广义表  $A=((x,(a,B)),(x,(a,B),y))$ , 则运算  $\text{head}(\text{head}(\text{tail}(A)))$  的结果为 ( )。

- A.x    B.(a,B)    C.(x,(a,B))    D.A

8. 下列广义表用图来表示时, 分支结点最多的是 ( )。

- A.L=((x,(a,B)),(x,(a,B),y))    B.A=(s,(a,B))  
C.B=((x,(a,B),y))    D.D=((a,B),(c,(a,B),D))

9. 通常对数组进行的两种基本操作是 ( )。

- A.建立与删除    B.索引和修改  
C.查找和修改    D.查找与索引

10. 假定在数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到 10, 从首地址

SA 开始连续存放在存储器内, 存放该数组至少需要的单元数为 ( )。

- A.80    B.100    C.240    D.270

11. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 8, 列下标 j 从 1 到 10, 从首地址 SA 开始连续存放在存储器内, 该数组按行存放时, 元素  $A[8][5]$  的起始地址为 ( )。

- A.SA+141    B.SA+144    C.SA+222    D.SA+225

12. 稀疏矩阵一般的压缩存储方法有两种, 即 ( )。

- A.二维数组和三维数组    B.三元组和散列  
C.三元组和十字链表    D.散列和十字链表

13. 若采用三元组压缩技术存储稀疏矩阵, 只要把每个元素的行下标和列下标互换, 就完成了对该矩阵的转置运算, 这种观点 ( )。

- A.正确    B.不正确

14. 一个广义表的表头总是一个 ( )。

- A.广义表    B.元素    C.空表    D.元素或广义表

15. 一个广义表的表尾总是一个 ( )。

A.广义表 B.元素 C.空表 D.元素或广义表

16. 数组就是矩阵，矩阵就是数组，这种说法（）。

A.正确 B.错误

C.前句对，后句错 D.后句对

## 二、填空题

1. 一维数组的逻辑结构是\_\_\_\_\_，存储结构是\_\_\_\_\_；对于二维或多维数组，分为\_\_\_\_\_和\_\_\_\_\_两种不同的存储方式。

2. 对于一个二维数组  $A[m][n]$ ，若按行序为主序存储，则任一元素  $A[i][j]$  相对于  $A[0][0]$  的地址为\_\_\_\_\_。

3. 一个广义表为  $(a, (a, b), d, e, ((i, j), k))$ ，则该广义表的长度为\_\_\_\_，深度为\_\_\_\_。

4. 一个稀疏矩阵为\_\_\_\_\_，则对应的三元组线性表为\_\_\_\_\_。

5. 一个  $n \times n$  的对称矩阵，如果以行为主序或以列为主序存入内存，则其容量为\_\_\_\_\_。

6. 已知广义表  $A=((a,b,c),(d,e,f))$ ，则运算  $\text{head}(\text{tail}(\text{tail}(A)))=$ \_\_\_\_\_。

7. 设有一个 10 阶的对称矩阵  $A$ ，采用压缩存储方式以行序为主序存储， $a$  为第一个元素，其存储地址为 0，每个元素占有 1 个存储地址空间，则  $a$  的地址为\_\_\_\_\_。

8. 已知广义表  $Ls=(a,(b,c,d),e)$ ，运用  $\text{head}$  和  $\text{tail}$  函数取出  $Ls$  中的原子  $b$  的运算是\_\_\_\_\_。

9. 三维数组  $R[c_1 \dots d_1, c_2 \dots d_2, c_3 \dots d_3]$  共含有\_\_\_\_\_个元素。（其中： $c_1 \leq d_1, c_2 \leq d_2, c_3 \leq d_3$ ）

10. 数组  $A[1 \dots 10, -2 \dots 6, 2 \dots 8]$  以行优先的顺序存储，设第一个元素的首地址是 100，每个元素占 3 个存储长度的存储空间，则元素  $A[5, 0, 7]$  的存储地址为\_\_\_\_\_。

## 三、判断题

1. 数组可看作基本线性表的一种推广，因此与线性表一样，可以对它进行插入、删除等操作。（）

2. 多维数组可以看作数据元素也是基本线性表的基本线性表。（）

3. 以行为主序或以列为主序对于多维数组的存储没有影响。（）

4. 对于不同的特殊矩阵应该采用不同的存储方式。（）

5. 采用压缩存储之后，下三角矩阵的存储空间可以节约一半。（）

6. 在一般情况下，采用压缩存储之后，对称矩阵是所有特殊矩阵中存储空间节约最多的。（）

7. 矩阵不仅是表示多维数组，而且是表示图的重要工具。（）

8. 距阵中的数据元素可以是不同的数据类型。（）

9. 矩阵中的行列数往往是不相等的。（）

10. 广义表的表头可以是广义表，也可以是单个元素。（）

11. 广义表的表尾一定是一个广义表。（）

12. 广义表的元素可以是子表，也可以是单元素。（）

13. 广义表不能递归定义。（）

14. 广义表实际上是基本线性表的推广。（）

15. 广义表的组成元素可以是不同形式的元素。（）

#### 习题 4 参考答案

##### 一、单项选择题

1. A 2. A 3. A 4. B 5. BA 6. C 7. A 8. A 9. C 10. C 11. C 12. C 13. B 14. D 15. A 16. B

##### 二、填空题

1. 线性结构，顺序结构，以行为主序，以列为主序
2.  $i \times n + j$  个元素位置
3. 5, 3
4.  $( (0, 2, 2), (1, 0, 3), (2, 2, -1), (2, 3, 5) )$
5.  $n \times (n+1)/2$
6. e
7. 41
8. `head(head(tail(Ls)))`
9.  $(d - c + 1) \times (d - c + 1) \times (d - c + 1)$
10. 913

##### 三、判断题

1. ✗ 2. ✓ 3. ✓ 4. ✓ 5. ✗ 6. ✗ 7. ✓ 8. ✗ 9. ✗ 10. ✓ 11. ✓ 12. ✓ 13. ✗ 14. ✓ 15. ✓

#### 习题 5

##### 一、单项选择题

1. 在一棵度为 3 的树中，度为 3 的结点数为 2 个，度为 2 的结点数为 1 个，度为 1 的结点数为 2 个，则度为 0 的结点数为（ ）个。  
A. 4 B. 5 C. 6 D. 7
2. 假设在一棵二叉树中，双分支结点数为 15，单分支结点数为 30 个，则叶子结点数为（ ）个。  
A. 15 B. 16 C. 17 D. 47
3. 假定一棵三叉树的结点数为 50，则它的最小高度为（ ）。  
A. 3 B. 4 C. 5 D. 6
4. 在一棵二叉树上第 4 层的结点数最多为（ ）。  
A. 2 B. 4 C. 6 D. 8
5. 用顺序存储的方法将完全二叉树中的所有结点逐层存放在数组中  $R[1..n]$ ，结点  $R[i]$  若有左孩子，其左孩子的编号为结点（ ）。  
A.  $R[2i+1]$  B.  $R[2i]$  C.  $R[i/2]$  D.  $R[2i-1]$
6. 由权值分别为 3, 8, 6, 2, 5 的叶子结点生成一棵哈夫曼树，它的带权路径长度为（ ）。  
A. 24 B. 48 C. 72 D. 53
7. 线索二叉树是一种（ ）结构。  
A. 逻辑 B. 逻辑和存储 C. 物理 D. 线性
8. 线索二叉树中，结点  $p$  没有左子树的充要条件是（ ）。  
A.  $p->lc=NULL$  B.  $p->ltag=1$

- C.  $p->ltag=1$  且  $p->lc=NULL$  D. 以上都不对
9. 设  $n, m$  为一棵二叉树上的两个结点，在中序遍历序列中  $n$  在  $m$  前的条件是（ ）。
- A.  $n$  在  $m$  右方 B.  $n$  在  $m$  左方  
C.  $n$  是  $m$  的祖先 D.  $n$  是  $m$  的子孙
10. 如果  $F$  是由有序树  $T$  转换而来的二叉树，那么  $T$  中结点的前序就是  $F$  中结点的（ ）。
- A. 中序 B. 前序 C. 后序 D. 层次序
11. 欲实现任意二叉树的后序遍历的非递归算法而不必使用栈，最佳方案是二叉树采用（ ）存储结构。
- A. 三叉链表 B. 广义表 C. 二叉链表 D. 顺序
12. 下面叙述正确的是（ ）。
- A. 二叉树是特殊的树  
B. 二叉树等价于度为 2 的树  
C. 完全二叉树必为满二叉树  
D. 二叉树的左右子树有次序之分
13. 任何一棵二叉树的叶子结点在先序、中序和后序遍历序列中的相对次序（ ）。
- A. 不发生改变 B. 发生改变  
C. 不能确定 D. 以上都不对
14. 已知一棵完全二叉树的结点总数为 9 个，则最后一层的结点数为（ ）。
- A. 1 B. 2 C. 3 D. 4
15. 根据先序序列 ABDC 和中序序列 DBAC 确定对应的二叉树，该二叉树（ ）。
- A. 是完全二叉树 B. 不是完全二叉树  
C. 是满二叉树 D. 不是满二叉树
- 二、判断题**
1. 二叉树中每个结点的度不能超过 2，所以二叉树是一种特殊的树。 （ ）
2. 二叉树的前序遍历中，任意结点均处在其子女结点之前。 （ ）
3. 线索二叉树是一种逻辑结构。 （ ）
4. 哈夫曼树的总结点个数（多于 1 时）不能为偶数。 （ ）
5. 由二叉树的先序序列和后序序列可以唯一确定一颗二叉树。 （ ）
6. 树的后序遍历与其对应的二叉树的后序遍历序列相同。 （ ）
7. 根据任意一种遍历序列即可唯一确定对应的二叉树。 （ ）
8. 满二叉树也是完全二叉树。 （ ）
9. 哈夫曼树一定是完全二叉树。 （ ）
10. 树的子树是无序的。 （ ）

**三、填空题**

1. 假定一棵树的广义表表示为 A (B (E), C (F (H, I, J), G), D)，则该树的度为\_\_\_\_\_，树的深度为\_\_\_\_\_，终端结点的个数为\_\_\_\_\_，单分支结点的个数为\_\_\_\_\_，双分支结点的个数为\_\_\_\_\_，三分支结点的个数为\_\_\_\_\_，C 结点的双亲结点为\_\_\_\_\_，其孩子结点为\_\_\_\_\_和\_\_\_\_\_结点。
2. 设  $F$  是一个森林， $B$  是由  $F$  转换得到的二叉树， $F$  中有  $n$  个非终端结点，则  $B$  中右指针域为空的结点

有\_\_\_\_\_个。

3. 对于一个有  $n$  个结点的二叉树, 当它为一棵\_\_\_\_\_二叉树时具有最小高度, 即为\_\_\_\_\_, 当它为一棵单支树具有\_\_\_\_\_高度, 即为\_\_\_\_\_。
4. 由带权为 3, 9, 6, 2, 5 的 5 个叶子结点构成一棵哈夫曼树, 则带权路径长度为\_\_\_\_\_。
5. 在一棵二叉排序树上按\_\_\_\_\_遍历得到的结点序列是一个有序序列。
6. 对于一棵具有  $n$  个结点的二叉树, 当进行链接存储时, 其二叉链表中的指针域的总数为\_\_\_\_\_个, 其中\_\_\_\_\_个用于链接孩子结点, \_\_\_\_\_个空闲着。
7. 在一棵二叉树中, 度为 0 的结点个数为  $n_0$ , 度为 2 的结点个数为  $n_2$ , 则  $n_0 = \underline{\hspace{2cm}}$ 。
8. 一棵深度为  $k$  的满二叉树的结点总数为\_\_\_\_\_, 一棵深度为  $k$  的完全二叉树的结点总数的最小值为\_\_\_\_\_, 最大值为\_\_\_\_\_。
9. 由三个结点构成的二叉树, 共有\_\_\_\_\_种不同的形态。
10. 设高度为  $h$  的二叉树中只有度为 0 和度为 2 的结点, 则此类二叉树中所包含的结点数至少为\_\_\_\_\_。
11. 一棵含有  $n$  个结点的  $k$  叉树, \_\_\_\_\_形态达到最大深度, \_\_\_\_\_形态达到最小深度。
12. 对于一棵具有  $n$  个结点的二叉树, 若一个结点的编号为  $i (1 \leq i \leq n)$ , 则它的左孩子结点的编号为\_\_\_\_\_, 右孩子结点的编号为\_\_\_\_\_, 双亲结点的编号为\_\_\_\_\_。
13. 对于一棵具有  $n$  个结点的二叉树, 采用二叉链表存储时, 链表中指针域的总数为\_\_\_\_\_个, 其中\_\_\_\_\_个用于链接孩子结点, \_\_\_\_\_个空闲着。
14. 哈夫曼树是指\_\_\_\_\_的二叉树。
15. 空树是指\_\_\_\_\_, 最小的树是指\_\_\_\_\_。
16. 二叉树的链式存储结构有\_\_\_\_\_和\_\_\_\_\_两种。
17. 三叉链表比二叉链表多一个指向\_\_\_\_\_的指针域。
18. 线索是指\_\_\_\_\_。
19. 线索链表中的  $rtag$  域值为\_\_\_\_\_时, 表示该结点无右孩子, 此时\_\_\_\_\_域为指向该结点后继线索的指针。
20. 本节中我们学习的树的存储结构有\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

#### 四、应用题

1. 已知一棵树边的集合为

$\{<i, m>, <i, n>, <e, i>, <b, e>, <b, d>, <a, b>, <g, j>, <g, k>, <c, g>, <c, f>, <h, l>, <c, h>, <a, c>\}$ , 请画出这棵树, 并回答下列问题:

- (1) 哪个是根结点?
- (2) 哪些是叶子结点?
- (3) 哪个是结点  $g$  的双亲?
- (4) 哪些是结点  $g$  的祖先?
- (5) 哪些是结点  $g$  的孩子?
- (6) 哪些是结点  $e$  的孩子?
- (7) 哪些是结点  $e$  的兄弟? 哪些是结点  $f$  的兄弟?
- (8) 结点  $b$  和  $n$  的层次号分别是什么?
- (9) 树的深度是多少?

- (10) 以结点 c 为根的子树深度是多少?
2. 一棵度为 2 的树与一棵二叉树有何区别。
3. 试分别画出具有 3 个结点的树和二叉树的所有不同形态?
4. 已知用一维数组存放的一棵完全二叉树: ABCDEFGHIJKL, 写出该二叉树的先序、中序和后序遍历序列。
5. 一棵深度为 H 的满 k 叉树有如下性质: 第 H 层上的结点都是叶子结点, 其余各层上每个结点都有 k 棵非空子树, 如果按层次自上至下, 从左到右顺序从 1 开始对全部结点编号, 回答下列问题:
- (1) 各层的结点数目是多少?
  - (2) 编号为 n 的结点的父结点如果存在, 编号是多少?
  - (3) 编号为 n 的结点的第 i 个孩子结点如果存在, 编号是多少?
  - (4) 编号为 n 的结点有右兄弟的条件是什么? 其右兄弟的编号是多少?
6. 找出所有满足下列条件的二叉树:
- (1) 它们在先序遍历和中序遍历时, 得到的遍历序列相同;
  - (2) 它们在后序遍历和中序遍历时, 得到的遍历序列相同;
  - (3) 它们在先序遍历和后序遍历时, 得到的遍历序列相同;
7. 假设一棵二叉树的先序序列为 EBADCFHGIJK, 中序序列为 ABCDEFGHIJK, 请写出该二叉树的后序遍历序列。
8. 假设一棵二叉树的后序序列为 DCEGBFHKJIA, 中序序列为 DCBGAEHFIJK, 请写出该二叉树的后序遍历序列。
9. 给出如图 5-14 所示的森林的先根、后根遍历结点序列, 然后画出该森林对应的二叉树。
10. 给定一组权值 (5, 9, 11, 2, 7, 16), 试设计相应的哈夫曼树。

## 五、算法设计题

1. 一棵具有 n 个结点的完全二叉树以一维数组作为存储结构, 试设计一个对该完全二叉树进行先序遍历的算法。
2. 给定一棵用二叉链表表示的二叉树, 其中的指针 t 指向根结点, 试写出从根开始, 按层次遍历二叉树的算法, 同层的结点按从左至右的次序访问。
3. 写出在中序线索二叉树中结点 P 的右子树中插入一个结点 s 的算法。
4. 给定一棵二叉树, 用二叉链表表示, 其根指针为 t, 试写出求该二叉树中结点 n 的双亲结点的算法。若没有结点 n 或者该结点没有双亲结点, 分别输出相应的信息; 若结点 n 有双亲, 输出其双亲的值。

## 习题 5 参考答案

### 一、单项选择题

1. C 2. B 3. C 4. D 5. B 6. D 7. C 8. B 9. B 10. B 11. A 12. D 13. A 14. B 15. A

## 二、判断题

1. × 2. √ 3. × 4. √ 5. × 6. √ 7. √ 8. √ 9. × 10. ×

## 三、填空题

1. 3, 4, 6, 1, 1, 2, A, F, G

2. n+1

3. 完全, , 最大, n

4. 55

5. 中序

6. 2n, n-1, n+1

7. n<sup>2</sup>+1

8. 2k-1, 2k-1, 2k-1

9. 5

10. 2h-1

11. 单支树, 完全二叉树

12. 2i, 2i+1, i/2 (或?i/2?)

13. 2n, n-1, n+1

14. 带权路径长度最小

15. 结点数为 0, 只有一个根结点的树

16. 二叉链表, 三叉链表

17. 双亲结点

18. 指向结点前驱和后继信息的指针

19. 1, RChild

20. 孩子表示法, 双亲表示法, 长子兄弟表示法

## 四、应用题

1. 解答:

根据给定的边确定的树如图 5-15 所示。

其中根结点为 a;

叶子结点有: d、m、n、j、k、f、l;

c 是结点 g 的双亲;

a、c 是结点 g 的祖先;

j、k 是结点 g 的孩子;

m、n 是结点 e 的子孙;

e 是结点 d 的兄弟;

g、h 是结点 f 的兄弟;

结点 b 和 n 的层次号分别是 2 和 5;

树的深度为 5。

2. 解答:

度为 2 的树有两个分支，但分支没有左右之分；一棵二叉树也有两个分支，但有左右之分，左右子树不能交换。

3. 解答：略

4. 解答：

先序序列：ABDHIEJKCFLG

中序序列：HDIBJEKALFCG

后序序列：HIDJKEBLFGCA

5. 解答：

- (1) 第  $i$  层上的结点数目是  $m^{i-1}$ 。
- (2) 编号为  $n$  的结点的父结点如果存在，编号是  $((n-2)/m)+1$ 。
- (3) 编号为  $n$  的结点的第  $i$  个孩子结点如果存在，编号是  $(n-1)*m+i+1$ 。
- (4) 编号为  $n$  的结点有右兄弟的条件是  $(n-1)%m \neq 0$ 。其右兄弟的编号是  $n+1$ 。

6. 解答：

- (1) 先序序列和中序序列相同的二叉树为：空树或者任一结点均无左孩子的非空二叉树；
- (2) 中序序列和后序序列相同的二叉树为：空树或者任一结点均无右孩子的非空二叉树；
- (3) 先序序列和后序序列相同的二叉树为：空树或仅有一个结点的二叉树。

7. 解答：后序序列：ACDBGJKIHFE

8. 解答：先序序列：ABCDGEIHFJK

9. 解答：

先根遍历：ABCDEFGHIJKLMNO

后根遍历：BDEFCAHJIGKNOML

森林转换成二叉树如图 5-16 所示。

10. 解答：构造而成的哈夫曼树如图 5-17 所示。

## 试题及答案

### 一、单选题（每题 2 分，共 20 分）

1. 1. 对一个算法的评价，不包括如下（B）方面的内容。  
A. 健壮性和可读性 B. 并行性 C. 正确性 D. 时空复杂度
2. 在带有头结点的单链表 HL 中，要向表头插入一个由指针 p 指向的结点，则执行( )。  
A. p->next=HL->next; HL->next=p; B. p->next=HL; HL=p;  
C. p->next=HL; p=HL; D. HL=p; p->next=HL;
3. 3. 对线性表，在下列哪种情况下应当采用链表表示？（ ）  
A. 经常需要随机地存取元素 B. 经常需要进行插入和删除操作  
C. 表中元素需要占据一片连续的存储空间 D. 表中元素的个数不变
4. 4. 一个栈的输入序列为 1 2 3，则下列序列中不可能是栈的输出序列的是（C）  
A. 2 3 1 B. 3 2 1 C. 3 1 2 D. 1 2 3
5. 5. AOV 网是一种（ ）。  
A. 有向图 B. 无向图 C. 无向无环图 D. 有向无环图
6. 6. 采用开放定址法处理散列表的冲突时，其平均查找长度（ ）。  
A. 低于链接法处理冲突 B. 高于链接法处理冲突  
C. 与链接法处理冲突相同 D. 高于二分查找
7. 7. 若需要利用形参直接访问实参时，应将形参变量说明为（ ）参数。  
A. 值 B. 函数 C. 指针 D. 引用
8. 8. 在稀疏矩阵的带行指针向量的链接存储中，每个单链表中的结点都具有相同的（ ）。  
A. 行号 B. 列号 C. 元素值 D. 非零元素个数
9. 9. 快速排序在最坏情况下的时间复杂度为（ ）。  
A.  $O(\log_2 n)$  B.  $O(n \log_2 n)$  C.  $O(n)$  D.  $O(n^2)$
10. 10. 从二叉搜索树中查找一个元素时，其时间复杂度大致为（ ）。

- A. O(n)    B. O(1)    C. O(log<sub>2</sub>n)    D. O(n<sup>2</sup>)

- 二、运算题 (每题 6 分, 共 24 分)**
1. 数据结构是指数据及其相互之间的\_\_\_\_\_。当结点之间存在 M 对 N (M: N) 的联系时, 称这种结构为\_\_\_\_\_。
  2. 队列的插入操作是在队列的\_\_\_\_尾\_\_\_\_进行, 删除操作是在队列的\_\_\_\_首\_\_\_\_进行。
  3. 当用长度为 N 的数组顺序存储一个栈时, 假定用 top==N 表示栈空, 则表示栈满的条件是\_\_\_\_top==0\_\_\_\_ (要超出才为满)。
  4. 对于一个长度为 n 的单链存储的线性表, 在表头插入元素的时间复杂度为\_\_\_\_\_, 在表尾插入元素的时间复杂度为\_\_\_\_\_。
  5. 设 W 为一个二维数组, 其每个数据元素占用 4 个字节, 行下标 i 从 0 到 7, 列下标 j 从 0 到 3, 则二维数组 W 的数据元素共占用\_\_\_\_\_个字节。W 中第 6 行的元素和第 4 列的元素共占用\_\_\_\_\_个字节。若按行顺序存放二维数组 W, 其起始地址为 100, 则二维数组元素 W[6, 3]的起始地址为\_\_\_\_\_。
  6. 广义表 A=(a,(a,b),(a,b),c), 则它的深度为\_\_\_\_\_, 它的长度为\_\_\_\_\_。
  7. 二叉树是指度为 2 的\_\_\_\_\_树。一棵结点数为 N 的二叉树, 其所有结点的度的总和是\_\_\_\_\_。
  8. 对一棵二叉搜索树进行中序遍历时, 得到的结点序列是一个\_\_\_\_\_。对一棵由算术表达式组成的二叉语法树进行后序遍历得到的结点序列是该算术表达式的\_\_\_\_\_。
  9. 对于一棵具有 n 个结点的二叉树, 用二叉链表存储时, 其指针总数为\_\_\_\_\_个, 其中\_\_\_\_\_个用于指向孩子, \_\_\_\_\_个指针是空闲的。
  10. 若对一棵完全二叉树从 0 开始进行结点的编号, 并按此编号把它顺序存储到一维数组 A 中, 即编号为 0 的结点存储到 A[0]中。其余类推, 则 A[i]元素的左孩子元素为\_\_\_\_\_, 右孩子元素为\_\_\_\_\_, 双亲元素为\_\_\_\_\_。
  11. 在线性表的散列存储中, 处理冲突的常用方法有\_\_\_\_\_和\_\_\_\_\_两种。
  12. 当待排序的记录数较大, 排序码较随机且对稳定性不作要求时, 宜采用\_\_\_\_\_排序; 当待排序的记录数较大, 存储空间允许且要求排序是稳定时, 宜采用\_\_\_\_\_排序。

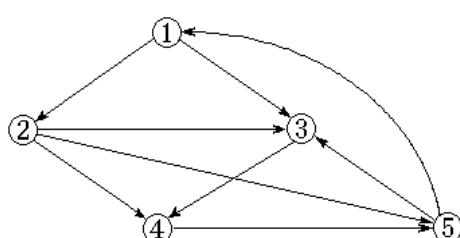
**三、运算题 (每题 6 分, 共 24 分)**

1. 已知一个 6×5 稀疏矩阵如下所示,

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 \\ 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 \end{bmatrix}$$

试:

- (1) (1) 写出它的三元组线性表;  
 (2) (2) 给出三元组线性表的顺序存储表示。
2. 设有一个输入数据的序列是 { 46, 25, 78, 62, 12, 80 }, 试画出从空树起, 逐个输入各个数据而生成的二叉搜索树。
3. 对于图 6 所示的有向图若存储它采用邻接表, 并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的, 试写出:  
 (1) 从顶点①出发进行深度优先搜索所得到的深度优先生成树;



- (2) 从顶点②出发进行广度优先搜索所得到的广度优先生成树;
4. 已知一个图的顶点集 V 和边集 E 分别为:  
 $V=\{1,2,3,4,5,6,7\};$

足

图 6

E={<2,1>,<3,2>,<3,6>,<4,3>,<4,5>,<4,6>,<5,1>,<5,7>,<6,1>,<6,2>,<6,5>};  
若存储它采用邻接表，并且每个顶点邻接表中的边结点都是按照终点序号从小到大的次序链接的，按主教材中介绍的拓扑排序算法进行排序，试给出得到的拓扑排序的序列。

#### 四、阅读算法（每题 7 分，共 14 分）

1. 1. int Prime(int n)  
 {  
 int i=1;  
 int x=(int) sqrt(n);  
 while (++i<=x)  
 if (n%i==0) break;  
 if (i>x) return 1;  
 else return 0;  
 }  
 (1) (1) 指出该算法的功能；  
 (2) (2) 该算法的时间复杂度是多少？  
 2. 2. 写出下述算法的功能：  
 void AJ(adjlist GL, int i, int n)  
 {  
 Queue Q;  
 InitQueue(Q);  
 cout<<i<<' ';  
 visited[i]=true;  
 QInsert(Q,i);  
 while(!QueueEmpty(Q)) {  
 int k=QDelete(Q);  
 edgenode\* p=GL[k];  
 while(p!=NULL)  
 {  
 int j=p->adjvex;  
 if(!visited[j])  
 {  
 cout<<j<<' ';  
 visited[j]=true;  
 QInsert(Q,j);  
 }  
 p=p->next;  
 }  
 }  
 }

#### 五、算法填空（共 8 分）

如下为二分查找的非递归算法，试将其填写完整。

Int Binsch(ElemType A[ ],int n,KeyType K)  
 {  
 int low=0;  
 int high=n-1;  
 while (low<=high)  
 {  
 int mid=\_\_\_\_\_;  
 if (K==A[mid].key) return mid; //查找成功，返回元素的下标  
 else if (K<[mid].key)  
 \_\_\_\_\_; //在左子表上继续查找  
 else \_\_\_\_\_; //在右子表上继续查找  
 }  
 return -1; //查找失败，返回-1  
 }

#### 六、编写算法（共 8 分）

HL 是单链表的头指针，试写出删除头结点的算法。  
 ElemtType DeleFront(LNode \* & HL)

参考答案

#### 一、单选题（每题 2 分，共 20 分）

1.B 2.A 3.B 4.C 5.D 6.B 7.D 8.A 9.D 10.C

**二、填空题 (每空 1 分, 共 26 分)**

1. 联系图 (或图结构)
2. 尾首
3.  $\text{top} = 0$
4.  $O(1) O(n)$
5. 128 44 108
6. 3 3

6	5	5
1	5	1
3	2	-1
4	5	-2
5	1	5
6	3	7

图 7

7. 有序  $n-1$
8. 有序序列 后缀表达式 (或逆波兰式)
9.  $2n \ n-1 \ n+1$
10.  $2i+1 \ 2i+2 \ (i-1)/2$
11. 开放定址法 链接法
12. 快速 归并

**三、运算题 (每题 6 分, 共 24 分)**

1. (1)  $((1,5,1),(3,2,-1),(4,5,-2),(5,1,5),(6,3,7))$  (3 分)
- (2) 三元组线性表的顺序存储表示如图 7 示。

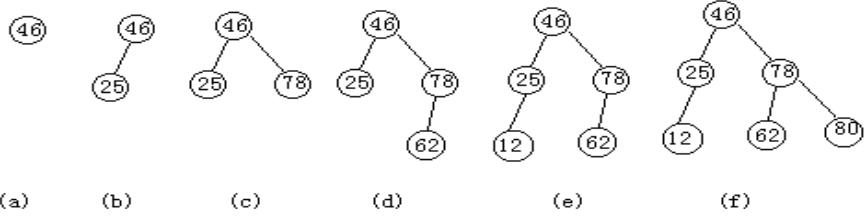


图 8

2. 如图 8 所示。
3. DFS: ①②③④⑤

BFS: ①②③④⑤

4. 拓朴排序为: 4 3 6 5 7 2 1

**四、阅读算法 (每题 7 分, 共 14 分)**

1. (1) 判断  $n$  是否是素数 (或质数)

(2)  $O(\sqrt{n})$

2. 功能为: 从初始点  $v_i$  出发广度优先搜索由邻接表  $GL$  所表示的图。

**五、算法填空 (8 分)**

$(low+high)/2$        $high=mid-1$        $low=mid+1$

**六、编写算法 (8 分)**

`ElemType DeleteFront(LNode * & HL)`

```
{
    if (HL==NULL){
        cerr<<"空表"<<endl;
        exit(1);
    }
    LNode* p=HL;
    HL=HL->next;
    ElemType temp=p->data;
    delete p;
    return temp;
}
```

**一、单选题 (每题 2 分, 共 20 分)**

1. 1. 栈和队列的共同特点是( )。

- A. 只允许在端点处插入和删除元素
- B. 都是先进后出
- C. 都是先进先出
- D. 没有共同点

2. 2. 用链接方式存储的队列, 在进行插入运算时( ).

- A. 仅修改头指针
- B. 头、尾指针都要修改

- C. 仅修改尾指针      D. 头、尾指针可能都要修改
3. 3. 以下数据结构中哪一个是非线性结构? ( )  
A. 队列      B. 栈      C. 线性表      D. 二叉树
4. 4. 设有一个二维数组  $A[m][n]$ , 假设  $A[0][0]$  存放位置在  $644_{(10)}$ ,  $A[2][2]$  存放位置在  $676_{(10)}$ , 每个元素占一个空间, 问  $A[3][3]_{(10)}$  存放在什么位置? 脚注<sub>(10)</sub>表示用 10 进制表示。  
A. 688      B. 678      C. 692      D. 696
5. 5. 树最适合用来表示( )。  
A. 有序数据元素      B. 无序数据元素  
C. 元素之间具有分支层次关系的数据      D. 元素之间无联系的数据
6. 6. 二叉树的第 k 层的结点数最多为( )。  
A.  $2^{k-1}$       B.  $2K+1$       C.  $2K-1$       D.  $2^{k-1}$
7. 7. 若有 18 个元素的有序表存放在一维数组  $A[19]$  中, 第一个元素放  $A[1]$  中, 现进行二分查找, 则查找  $A[3]$  的比较序列的下标依次为( )  
A. 1, 2, 3      B. 9, 5, 2, 3  
C. 9, 5, 3      D. 9, 4, 2, 3
8. 8. 对 n 个记录的文件进行快速排序, 所需要的辅助存储空间大致为  
A.  $O(1)$       B.  $O(n)$       C.  $O(\log_2 n)$       D.  $O(n^2)$
9. 9. 对于线性表 (7, 34, 55, 25, 64, 46, 20, 10) 进行散列存储时, 若选用  $H(K) = K \% 9$  作为散列函数, 则散列地址为 1 的元素有( )个。  
A. 1      B. 2      C. 3      D. 4
10. 10. 设有 6 个结点的无向图, 该图至少应有( )条边才能确保是一个连通图。  
A. 5      B. 6      C. 7      D. 8

## 二、填空题 (每空 1 分, 共 26 分)

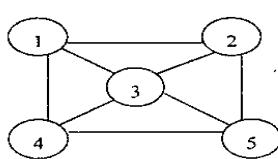
1. 通常从四个方面评价算法的质量: \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 一个算法的时间复杂度为  $(n^3 + n^2 \log_2 n + 14n)/n^2$ , 其数量级表示为\_\_\_\_\_。
3. 假定一棵树的广义表表示为  $A(C, D(E, F, G), H(I, J))$ , 则树中所含的结点数为\_\_\_\_\_个, 树的深度为\_\_\_\_\_, 树的度数为\_\_\_\_\_。
4. 4. 后缀算式  $9\ 2\ 3\ +\ 10\ 2\ /\ -$  的值为\_\_\_\_\_. 中缀算式  $(3+4X)-2Y/3$  对应的后缀算式为\_\_\_\_\_。
5. 若用链表存储一棵二叉树时, 每个结点除数据域外, 还有指向左孩子和右孩子的两个指针。在这种存储结构中,  $n$  个结点的二叉树共有\_\_\_\_\_个指针域, 其中有\_\_\_\_\_个指针域是存放了地址, 有\_\_\_\_\_个指针是空指针。
6. 对于一个具有  $n$  个顶点和  $e$  条边的有向图和无向图, 在其对应的邻接表中, 所含边结点分别有\_\_\_\_\_个和\_\_\_\_\_个。
7. AOV 网是一种\_\_\_\_\_的图。
8. 在一个具有  $n$  个顶点的无向完全图中, 包含有\_\_\_\_\_条边, 在一个具有  $n$  个顶点的有向完全图中, 包含有\_\_\_\_\_条边。
9. 假定一个线性表为  $(12, 23, 74, 55, 63, 40)$ , 若按 Key % 4 条件进行划分, 使得同一余数的元素成为一个子表, 则得到的四个子表分别为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
10. 向一棵 B\_树插入元素的过程中, 若最终引起树根结点的分裂, 则新树比原树的高度\_\_\_\_\_。
11. 在堆排序的过程中, 对任一分支结点进行筛选运算的时间复杂度为\_\_\_\_\_, 整个堆排序过程的时间复杂度为\_\_\_\_\_。
12. 12. 在快速排序、堆排序、归并排序中, \_\_\_\_\_排序是稳定的。

## 三、运算题 (每题 6 分, 共 24 分)

1. 1. 在如下数组 A 中链接存储了一个线性表, 表头指针为  $A[0].next$ , 试写出该线性表。

A	0	1	2	3	4	5	6	7
data		60	50	78	90	34		40
next	3	5	7	2	0	4		1

2. 2. 请画出图 10 的邻接矩阵和邻接表。



3. 3. 已知一个图的顶点集 V 和边集 E 分别为:  
 $V=\{1,2,3,4,5,6,7\};$        $E=\{(1,2)3,(1,3)5,(1,4)8,(2,5)10,(2,3)6,$   
 $(3,4)15,$   
 $(3,5)12,(3,6)9,(4,6)4,(4,7)20,(5,6)18,(6,7)25\};$   
用克鲁斯卡尔算法得到最小生成树, 试写出在最小生成树中依次得到的各条边。
4. 4. 画出向小根堆中加入数据 4, 2, 5, 8, 3 时, 每加入一个数据后堆的变化。

图 10

## 四、阅读算法 (每题 7 分, 共 14 分)

1. 1. LinkList mynote(LinkList L)

```

{//L是不带头结点的单链表的头指针
if(L&&L->next){
    q=L; L=L->next; p=L;
S1:    while(p->next) p=p->next;
S2:    p->next=q; q->next=NULL;

}
return L;
}

```

请回答下列问题：

- (1) 说明语句 S1 的功能；
- (2) 说明语句组 S2 的功能；
- (3) 设链表表示的线性表为  $(a_1, a_2, \dots, a_n)$ ，写出算法执行后的返回值所表示的线性表。

```

2. 2. void ABC(BTNode * BT)
{
if BT {
    ABC (BT->left);
    ABC (BT->right);
    cout<<BT->data<<' ';
}
}

```

该算法的功能是：

#### 五、五、 算法填空（共 8 分）

二叉搜索树的查找——递归算法：

```

bool Find(BTree* BST,ElemType& item)
{
    if (BST==NULL)
        return false; //查找失败
    else {
        if (item==BST->data){
            item=BST->data;//查找成功
            return _____;
        }
        else if(item<BST->data)
            return Find(_____,item);
        else return Find(_____,item);
    }/if
}

```

#### 六、六、 编写算法（共 8 分）

统计出单链表 HL 中结点的值等于给定值 X 的结点数。

```
int CountX(LNode* HL,ElemType x)
```

#### 参考答案

一、单选题（每题 2 分，共 20 分）

1.A 2.D 3.D 4.C 5.C 6.D 7.D 8.C 9.D 10.A

二、填空题（每空 1 分，共 26 分）

1. 1. 正确性 易读性 强壮性 高效率

2. 2.  $O(n)$

3. 3. 9 3 3

4. 4. -1 3 4  $X^* + 2 Y^* 3 / -$

5. 5.  $2n$   $n-1$   $n+1$

6. 6. e  $2e$

7. 7. 有向无回路

8. 8.  $n(n-1)/2$   $n(n-1)$

9. 9. (12, 40) ( ) (74) (23,55, 63)

10. 10. 增加 1

11. 11.  $O(\log_2 n)$   $O(n \log_2 n)$

12. 12. 归并

三、三、 运算题（每题 6 分，共 24 分）

1. 1. 线性表为：(78, 50, 40, 60, 34, 90)

0	1	1	1	0
1	0	1	0	1
1	1	0	1	1
1	0	1	0	1
0	1	1	1	0

2. 2. 邻接矩阵:  
邻接表如图 11 所示:

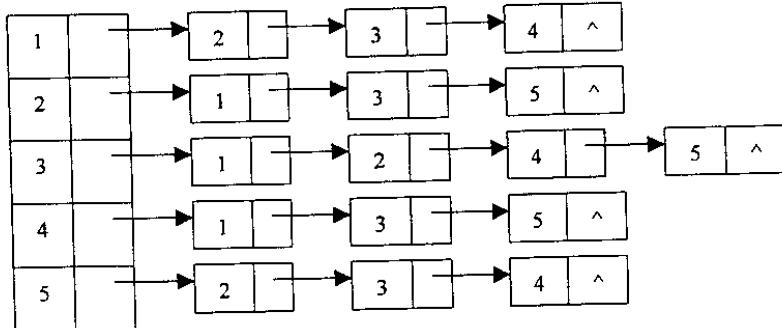


图 11

3. 3. 用克鲁斯卡尔算法得到的最小生成树为:  
(1,2)3, (4,6)4, (1,3)5, (1,4)8, (2,5)10, (4,7)20  
4. 4. 见图 12

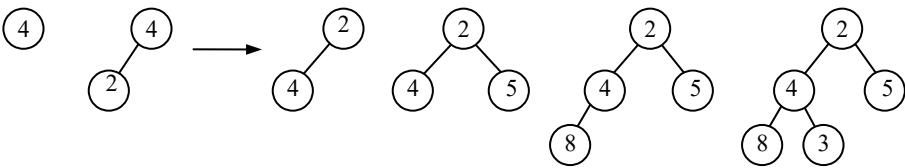


图 12

四、 四、 → 1. 1. (1) 查询链表的尾结点

(2) 将第一个结点链接到 8 的 4 作为新的尾结点

(3) 返回的线性表为  $(a_2, a_3, \dots, a_n, a_1)$

2. 2. 递归地后序遍历链式存储的二叉树。

五、 五、 → 算法填空 (每空 2 分, 共 8 分)

true BST->left BST->right

六、 六、 编写算法 (8 分)

```
int CountX(LNode* HL, ElemtType x)
{ int i=0; LNode* p=HL;/i 为计数器
  while(p!=NULL)
    { if (P->data==x) i++;
      p=p->next;
    } //while, 出循环时 i 中的值即为 x 结点个数
  return i;
}//CountX
```

一、 一、 单选题 (每小题 2 分, 共 8 分)

1. 1. 在一个长度为 n 的顺序线性表中顺序查找值为 x 的元素时, 查找成功时的平均查找长度 (即 x 与元素的平均比较次数, 假定查找每个元素的概率都相等) 为 ( )。

A n B n/2 C (n+1)/2 D (n-1)/2

1、 2、在一个单链表中,若 q 所指结点是 p 所指结点的前驱结点,若在 q 与 p 之间插入一个 s 所指的结点,则执行( )。

- A  $s \rightarrow \text{link} = p \rightarrow \text{link}$ ;  $p \rightarrow \text{link} = s$ ; B  $p \rightarrow \text{link} = s$ ;  $s \rightarrow \text{link} = q$ ;  
C  $p \rightarrow \text{link} = s \rightarrow \text{link}$ ;  $s \rightarrow \text{link} = p$ ; D  $q \rightarrow \text{link} = s$ ;  $s \rightarrow \text{link} = p$ ;

2、 3、 栈的插入和删除操作在( )进行。

- A 栈顶 B 栈底 C 任意位置 D 指定位置

3、 4、 由权值分别为 11, 8, 6, 2, 5 的叶子结点生成一棵哈夫曼树, 它的带权路径长度为( )

- A 24 B 71 C 48 D 53

二、 填空题(每空 1 分, 共 32 分)

1、 1、数据的逻辑结构被分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_四种。

2、 2、一种抽象数据类型包括\_\_\_\_\_和\_\_\_\_\_两个部分。

3、 3、在下面的数组 a 中链接存储着一个线性表, 表头指针为 a[0].next, 则该线性表为\_\_\_\_\_。

a	0	1	2	3	4	5	6	7	8
	60	56	42	38			74	25	
4	3	7	6	2			0	1	

data  
next

4、 4、在以 HL 为表头指针的带表头附加结点的单链表和循环单链表中, 判断链表为空的条件分别为\_\_\_\_\_和\_\_\_\_\_。

5、 5、用具有 n 个元素的一维数组存储一个循环队列, 则其队首指针总是指向队首元素的\_\_\_\_\_, 该循环队列的最大长度为\_\_\_\_\_。

6、 6、当堆栈采用顺序存储结构时, 栈顶元素的值可用\_\_\_\_\_表示; 当堆栈采用链接存储结构时, 栈顶元素的值可用\_\_\_\_\_表示。

7、 7、一棵高度为 5 的二叉树中最少含有\_\_\_\_\_个结点, 最多含有\_\_\_\_\_个结点;

一棵高度为 5 的理想平衡树中, 最少含有\_\_\_\_\_个结点, 最多含有\_\_\_\_\_个结点。

8、 8、在图的邻接表中, 每个结点被称为\_\_\_\_\_, 通常它包含三个域: 一是\_\_\_\_\_  
; 二是\_\_\_\_\_; 三是\_\_\_\_\_。

9、 9、在一个索引文件的索引表中, 每个索引项包含对应记录的\_\_\_\_\_和\_\_\_\_\_两项数据。

10、 10、假定一棵树的广义表表示为 A (B (C, D (E, F, G), H (I, J))), 则树中所含的结点数为\_\_\_\_\_个, 树的深度为\_\_\_\_\_, 树的度为\_\_\_\_\_, 结点 H 的双亲结点为\_\_\_\_\_, 孩子结点为\_\_\_\_\_。

11、 11、在堆排序的过程中, 对任一分支结点进行筛选运算的时间复杂度为\_\_\_\_\_, 整个堆排序过程的时间复杂度为\_\_\_\_\_。

12、 12、在对 m 阶的 B 树插入元素的过程中, 每向一个结点插入一个索引项(叶子结点中的索引项为关键字和空指针)后, 若该结点的索引项数等于\_\_\_\_\_个, 则必须把它分裂为\_\_\_\_\_个结点。

三、 三、 运算题(每小题 6 分, 共 24 分)

1、 1、已知一组记录的排序码为 (46, 79, 56, 38, 40, 80, 95, 24), 写出对其进行快速排序的每一次划分结果。

2、 2、一个线性表为 B=(12, 23, 45, 57, 20, 03, 78, 31, 15, 36), 设散列表为 HT[0..12], 散列函数为  $H(\text{key}) = \text{key} \% 13$  并用线性探查法解决冲突, 请画出散列表, 并计算等概率情况下查找成功的平均查找长度。

3、 3、已知一棵二叉树的前序遍历的结果序列是 ABECKFGHIJ, 中序遍历的结果是 EBCDAFHIGJ, 试写出这棵二叉树的后序遍历结果。

4、 4、已知一个图的顶点集 V 各边集 G 如下:

$V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ;  
 $E = \{(0, 1), (0, 4), (1, 2), (1, 7), (2, 8), (3, 4), (3, 8), (5, 6), (5, 8), (5, 9), (6, 7), (7, 8), (8, 9)\}$

当它用邻接矩阵表示和邻接表表示时, 分别写出从顶点  $V_0$  出发按深度优先搜索遍历得到的顶点序列和按广度优先搜索遍历等到的顶点序列。

假定每个顶点邻接表中的结点是按顶点序号从大到小的次序链接的。

图	深度优先序列	广度优先序列
邻接矩阵表示时		
邻接表表示时	第 39 页 共 73 页	

四、 四、 阅读算法，回答问题（每小题8分，共16分）

1、假定从键盘上输入一批整数，依次为：78 63 45 30 91 34 -1，请写出输出结果。

```
# include <iostream.h>
# include <stdlib.h>
const int stackmaxsize = 30;
typedef int elemtype;
struct stack {
    elemtype stack [stackmaxsize];
    int top;
};
# include "stack.h"
Void main ()
{
    stack a;
    initstack(a);
    int x;
    cin >>x;
    while (x! = -1) {
        push (a, x );
        cin >>x;
    }
    while (!stackempty (a))
        cout <<pop (a) <<"";
    cout <<endl;
}
```

该算法的输出结果为：

2、阅读以下二叉树操作算法，指出该算法的功能。

```
Template <class Type > void BinTree <Type> ::  
unknown (BinTreeNode<Type>*&t) {  
    BinTreeNode< Type> *p =t, *temp;  
    if (p!=NULL) {  
        temp = p->leftchild;  
        p->leftchild = p->rightchild;  
        p->rightchild = temp;  
        unknown(p->leftchild);  
        undnown(p->rightchild);  
    }  
}
```

该算法的功能是：\_\_\_\_\_

五、 五、 算法填空，在画有横线的地方填写合适的内容（10分）

对顺序存储的有序表进行二分查找的递归算法。

```
int Binsch( ElemtType A[ ],int low ,int high,KeyType K )  
{  
    if (low <= high)  
    {  
        int mid = 1  
        if ( K== A[ mid ].key )  
            return mid;  
        else if ( K < A[mid].key )  
            return 2  
        else  
            return 3  
    }  
    else  
        return 4
```

## 六、 六、 编写算法 (10 分)

编写算法，将一个结点类型为 Lnode 的单链表按逆序链接，即若原单链表中存储元素的次序为  $a_1, \dots, a_{n-1}, a_n$ ，则逆序链接后变为  $a_n, a_{n-1}, \dots, a_1$ 。  
 Void contrary (Lnode \* & HL)

## 数据结构试题 (答案)

### 一、 单选题 (每小题 2 分, 共 8 分)

题号	1	2	3	4
答案	C	D	A	B

### 二、 填空题 (每空 1 分, 共 32 分)

- 1: 集合、线性、树、图；
- 2: 数据描述、操作声明；
- 3: (38, 56, 25, 60, 42, 74)；
- 4: HL→next=NULL; HL=HL→next;
- 5: 前一个位置; n-1;
- 6: S.stack [S.top]; HS→data;
- 7: 5 31
- 8: 边结点、邻接点域、权域、链域；
- 9: 索引值域、开始位置域；
- 10: 10、3、3、B、I 和 J；
- 11:  $O(\log_2 n)$ 、 $O(n \log_2 n)$ ；
- 12: m、m - 1

### 三、 运算题 (每小题 6 分, 共 24 分)

1、

划分次序	划分结果
第一次	[38 24 40] 46 [56 80 95 79]
第二次	24 [38 40] 46 [56 80 95 79]
第三次	24 38 40 46 [56 80 95 79]
第四次	24 38 40 46 56 [80 95 79]
第五次	24 38 40 46 56 79 [80 95]
第六次	24 38 40 46 56 79 80 95

2、

0 1 2 3 4 5 6 7 8 9 10 11 12

78		15	03		57	45	20	31		23	36	12
----	--	----	----	--	----	----	----	----	--	----	----	----

查找成功的平均查找长度: ASL<sub>SUCC</sub>=14/10= 1.4

3、此二叉树的后序遍历结果是: EDCBIHJGFA

4、

图	深度优先序列	广度优先序列
邻接矩阵表示时	0, 1, 2, 8, 3, 4, 5, 6, 7, 9	0, 1, 4, 2, 7, 3, 8, 6, 5, 9
邻接表表示时	0, 4, 3, 8, 9, 5, 6, 7, 1, 2	0, 4, 1, 3, 7, 2, 8, 6, 9, 5

### 四、 阅读算法, 回答问题 (每小题 8 分, 共 16 分)

1、 1、 该算法的输入结果是: 34 91 30 45 63 78

2、 2、 该算法的功能是: 交换二叉树的左右子树的递归算法。

五、算法填空，在画有横线的地方填写合适的内容（10分）

- 1、1是： $(\text{low} + \text{high}) / 2;$
- 2是：Binsch(A,low,mid-1,K);
- 3是：Binsch(A,mid+1,high,K);
- 4是：-1;

六、编写算法（10分）

根据编程情况，酌情给分。

```
{  
    Lnode *P=HL;  
    HL=NULL;  
    While (p!=null)  
    {  
        Lnode*q=p;  
        P=p->next;  
        q->next=HL;  
        HL=q;  
    }  
}
```

### 第一部分 选择题(30分)

一、一、项选择题（本大题共15小题，每小题2分，共30分）在每小题列出的四个选项中只有一个选项是符合题目要求的，请将正确选项前的字母填在题后的括号内。

1. 算法指的是（ ）  
A. 计算机程序      B. 解决问题的计算方法  
C. 排序算法      D. 解决问题的有限运算序列
2. 线性表采用链式存储时，结点的存储地址（ ）  
A. 必须是不连续的  
B. 连续与否均可  
C. 必须是连续的  
D. 和头结点的存储地址相连续
3. 将长度为n的单链表链接在长度为m的单链表之后的算法的时间复杂度为（ ）  
A. O(1)      B. O(n)      C. O(m)      D. O(m+n)
4. 由两个栈共享一个向量空间的好处是：（ ）  
A. 减少存取时间，降低下溢发生的机率  
B. 节省存储空间，降低上溢发生的机率  
C. 减少存取时间，降低上溢发生的机率  
D. 节省存储空间，降低下溢发生的机率
5. 设数组 data[m]作为循环队列 SQ 的存储空间，front 为队头指针，rear 为队尾指针，则执行出队操作后其头指针 front 值为（ ）  
A. front=front+1      B. front=(front+1)%(m-1)  
C. front=(front-1)%m      D. front=(front+1)%m
6. 如下陈述中正确的是（ ）  
A. 串是一种特殊的线性表      B. 串的长度必须大于零  
C. 串中元素只能是字母      D. 空串就是空白串

7. 若目标串的长度为  $n$ , 模式串的长度为  $[n/3]$ , 则执行模式匹配算法时, 在最坏情况下的时间复杂度是( )

A.  $O(\frac{n}{3})$     B.  $O(n)$     C.  $O(n^2)$     D.  $O(n^3)$

8. 一个非空广义表的表头( )

- A. 不可能是子表    B. 只能是子表  
C. 只能是原子    D. 可以是子表或原子

9. 假设以带行标的三元组表表示稀疏矩阵, 则和下列行表

0	2	3	3	5
---	---	---	---	---

对应的稀疏矩阵是( )

A.  $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$     B.  $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$

C.  $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$     D.  $\begin{bmatrix} 0 & -8 & 0 & 6 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \\ -5 & 0 & 4 & 0 \\ 0 & 3 & 0 & 0 \end{bmatrix}$

10. 在一棵度为 3 的树中, 度为 3 的结点个数为 2, 度为 2 的结点个数为 1, 则度为 0 的结点个数为( )

- A. 4    B. 5    C. 6    D. 7

11. 在含  $n$  个顶点和  $e$  条边的无向图的邻接矩阵中, 零元素的个数为( )

- A.  $e$     B.  $2e$     C.  $n^2 - e$     D.  $n^2 - 2e$

12. 假设一个有  $n$  个顶点和  $e$  条弧的有向图用邻接表表示, 则删除与某个顶点  $v_i$  相关的所有弧的时间复杂度是( )

- A.  $O(n)$     B.  $O(e)$     C.  $O(n+e)$     D.  $O(n^*e)$

13. 用某种排序方法对关键字序列 (25, 84, 21, 47, 15, 27, 68, 35, 20) 进行排序时, 序列的变化情况如下:

20, 15, 21, 25, 47, 27, 68, 35, 84

15, 20, 21, 25, 35, 27, 47, 68, 84

15, 20, 21, 25, 27, 35, 47, 68, 84

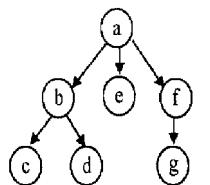
则所采用的排序方法是( )

- A. 选择排序    B. 希尔排序    C. 归并排序    D. 快速排序

14. 适于对动态查找表进行高效率查找的组织结构是 ( )  
 A. 有序表 B. 分块有序表 C. 三叉排序树 D. 线性链表
15. 不定长文件是指 ( )  
 A. 文件的长度不固定 B. 记录的长度不固定  
 C. 字段的长度不固定 D. 关键字项的长度不固定

### 第二部分 非选择题 (共 70 分)

- 二、填空题 (本大题共 10 小题, 每小题 2 分, 若有两个空格, 每个空格 1 分, 共 20 分) 不写解答过程,  
 将正确的答案写在每小题的空格内。错填或不填均无分。
16. 数据的逻辑结构是从逻辑关系上描述数据, 它与数据的\_\_\_\_\_无关, 是独立于计算机的。
17. 在一个带头结点的单循环链表中, p 指向尾结点的直接前驱, 则指向头结点的指针 head 可用 p 表示为 head=\_\_\_\_\_。
18. 栈顶的位置是随着\_\_\_\_\_操作而变化的。
19. 在串 S="structure" 中, 以 t 为首字符的子串有\_\_\_\_个。
20. 假设一个 9 阶的上三角矩阵 A 按列优先顺序压缩存储在一维数组 B 中, 其中 B[0] 存储矩阵中第 1 个元素 a<sub>1,1</sub>, 则 B[31] 中存放的元素是\_\_\_\_\_。
21. 已知一棵完全二叉树中共有 768 结点, 则该树中共有\_\_\_\_个叶子结点。



22. 已知一个图的广度优先生成树如右图所示, 则与此相对应的广度优先遍历序列为\_\_\_\_\_。

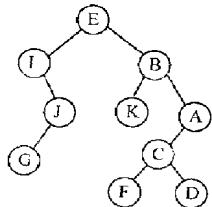
23. 在单链表上难以实现的排序方法有\_\_\_\_\_和\_\_\_\_\_。
24. 在有序表 (12, 24, 36, 48, 60, 72, 84) 中二分查找关键字 72 时所需进行的关键字比较次数为\_\_\_\_\_。
25. 多重表文件和倒排文件都归属于\_\_\_\_\_文件。

三、解答题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

26. 画出下列广义表的共享结构图形表示

$$P = ((z), (x, y), ((x, y), x), (z))$$

27. 请画出与下列二叉树对应的森林。



28 . 已知一个无向图的顶点集为 {a, b, c, d, e} ,其邻接矩阵如下所示

a	[
b	0 1 0 0 1
c	1 0 0 1 0
d	0 0 0 1 1
e	0 1 1 0 1
	1 0 1 1 0 ]

(1)画出该图的图形;

(2) 根据邻接矩阵从顶点 a 出发进行深度优先遍历和广度优先遍历, 写出相应的遍历序列。

29 . 已知一个散列表如下图所示:

		35		20			33		48			59
0	1	2	3	4	5	6	7	8	9	10	11	12

其散列函数为  $h(key)=key \% 13$ , 处理冲突的方法为双重散列法, 探查序列为:

$$h_i = (h(key) + i * h_1(key)) \% m \quad i=0,1,\dots, m-1$$

其中

$$h_1(key)=key \% 11+1$$

回答下列问题:

(1) 对表中关键字 35, 20, 33 和 48 进行查找时, 所需进行的比较次数各为多少?

(2) 该散列表在等概率查找时查找成功的平均查找长度为多少?

四、算法阅读题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

30 . 下列算法的功能是比较两个链串的大小, 其返回值为:

$$\text{comstr}(s_1, s_2) = \begin{cases} -1 & \text{当 } s_1 < s_2 \\ 0 & \text{当 } s_1 = s_2 \\ 1 & \text{当 } s_1 > s_2 \end{cases}$$

请在空白处填入适当的内容。

```
int comstr(LinkString s1, LinkString s2)
{//s1 和 s2 为两个链串的头指针
while(s1&&s2){
    if(s1->date<s2->date) return -1;
    if(s1->date>s2->date) return 1;
    ①;
    ②;
}
if(③) return -1;
if(④) return 1;
⑤;
```

```
}
```

```
①
```

```
②
```

```
③
```

```
④
```

```
⑤
```

31. 阅读下面的算法

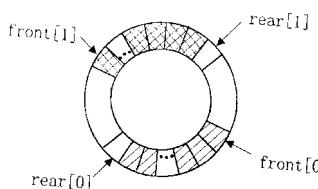
```
LinkList mynote(LinkList L)
{ //L是不带头结点的单链表的头指针
    if(L&&L->next){
        q=L; L=L->next; p=L;
        S1:   while(p->next) p=p->next;
        S2:   p->next=q; q->next=NULL;
    }
    return L;
}
```

请回答下列问题：

(1) 说明语句 S1 的功能；

(2) 说明语句组 S2 的功能；

(3) 设链表表示的线性表为  $(a_1, a_2, \dots, a_n)$ ，写出算法执行后的返回值所表示的线性表。



32. 假设两个队列共享一个循环向量空间（参见右下图），

其类型 Queue2 定义如下：

```
typedef struct{
    DateType data[MaxSize];
    int front[2],rear[2];
}Queue2;
```

对于  $i=0$  或  $1$ ， $\text{front}[i]$  和  $\text{rear}[i]$  分别为第  $i$  个队列的头指针和尾指针。请对以下算法填空，实现第  $i$  个队列的入队操作。

```
int EnQueue (Queue2*Q,int i,DateType x)
{//若第 i 个队列不满，则元素 x 入队列，并返回 1；否则返回 0
    if(i<0||i>1) return 0;
    if(Q->rear[i]==Q->front[_①_]) return 0;
    Q->data[_②_]=x;
    Q->rear[i]=_③_;
    return 1;
}
```

①

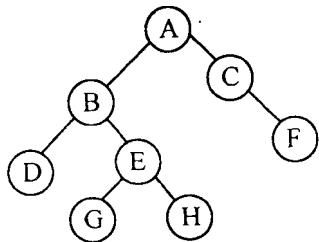
②

③

33. 已知二叉树的存储结构为二叉链表，阅读下面算法。

```
typedef struct node {  
    DateType data;  
    Struct node * next;  
}ListNode;  
typedef ListNode * LinkList ;  
LinkList Leafhead=NULL;  
Void Inorder (BinTree T)  
{  
    LinkList s;  
    If(T){  
        Inorder(T->lchild);  
        If ((!T->lchild)&&(!T->rchild)){  
            s=(ListNode*)malloc(sizeof(ListNode));  
            s->data=T->data;  
            s->next=Leafhead;  
            Leafhead=s;  
        }  
        Inorder(T->rchild);  
    }  
}
```

对于如下所示的二叉树



(1) 画出执行上述算法后所建立的结构；

(2) 说明该算法的功能。

五、算法设计题（本题共 10 分）

34. 阅读下列函数 arrange()

```

int arrange(int a[],int l,int h,int x)
{//l 和 h 分别为数据区的下界和上界
int i,j,t;
i=l; j=h;
while(i<j){
    while(i<j && a[j]>=x)j--;
    while(i<j && a[j]>=x)i++;
    if(i<j)
        { t=a[j]; a[j]=a[i]; a[i]=t; }
}
if(a[i]<x) return i;
else return i - 1;
}

```

- (1) 写出该函数的功能；  
(2) 写一个调用上述函数实现下列功能的算法：对一整型数组 b[n]中的元素进行重新排列，将所有负数均调整到数组的低下标端，将所有正数均调整到数组的高下标端，若有零值，则置于两者之间，并返回数组中零元素的个数。

### 数据结构试题参考答案

#### 一、 单项选择题（本大题共 15 小题，每小题 2 分，共 30 分）

- |       |        |        |        |           |
|-------|--------|--------|--------|-----------|
| 1 . D | 2 . B  | 3 . C  | 4 . B  | 5 . D     |
|       |        | 6 . A  | 7 . C  | 8 , D 9 . |
| A     | 10 . C |        |        | 11 .      |
| D     | 12 . C | 13 . D | 14 . C | 15 . B    |

#### 二、 填空题（本大题共 10 小题，每小题 2 分，共 20 分）

- 16 . 存储（或存储结构）  
17.p -> next -> next 18 .  
进栈和退栈 19 . 12  
20 . a<sub>4,8</sub>  
21 . 384  
22 . abefcdg  
23 . 快速排序、堆排序、希尔排序  
24 . 2 25. 多关键字

三、解答题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

26 .

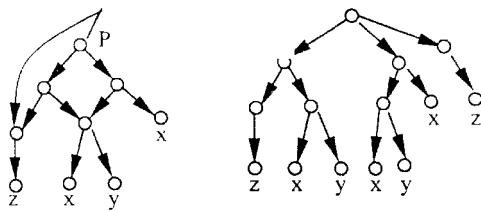
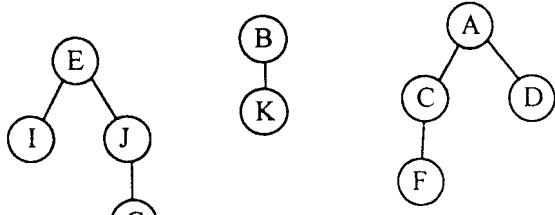


图 1

图 2

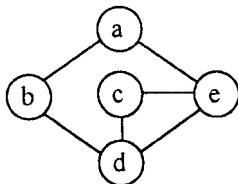
27 .



28 .

该图的图形

为:



深度优先遍历序列为: abdce

广度优先遍历序列为: abedc

29 . (1) 对关键字 35、20、33 和 48 进行查找的比较次数为 3、2、1、1；

$$\text{ASL} = \frac{3+2+1+1+2}{5} = \frac{9}{5}$$

(2) 平均查找长度

四、算法阅读题 (本大题共 4 小题, 每小题 5 分, 共 20 分)

30 . ① S1=S1->next

② s2=s2->next

③ s2(或 s2!=NULL 或 s2&&!s1)

④ s1(或 s1!=NULL 或 s1&&!s2)

⑤ return 0

31. (1) 查询链表的尾结点

(2) 将第一个结点链接到链表的尾部, 作为新的尾结点

(3) 返回的线性表为  $(a_2, a_3, \dots, a_n, a_1)$

32. ①(i + 1)%2(或 1 - i)

② Q->rear[i]

③( $Q ->rear[i] + ) \% Maxsize$



- (2) 中序遍历二叉树，按遍历序列中叶子结点数据域的值构建一个以 Leafhead 为头指针的逆序单链表（或按二叉树中叶子结点数据自右至左链接成一个链表）。

### 五、算法设计题 (本题共 10 分)

34. (1) 该函数的功能是：调整整数数组  $a[]$  中的元素并返回分界值  $i$ ，使所有  $< x$  的元素均落在  $a[1..i]$  上，使所有  $\geq x$  的元素均落在  $a[i+1..h]$  上。

(2) int f(int b[],int n) 或 int f(int b[],int n)

```

{           }

int p,q;           int p,q;

p=arrange(b,0,n - 1,0);    p=arrange(b,0,n - 1,1);

q= arrange(b,p+1,n - 1,1); q= arrange(b,0,p,0);

return q - p;           return p - q;

}

}

```

### 一、选择题 (20分)



## 二、填空题(30分)

1. 设顺序循环队列  $Q[0: m-1]$  的队头指针和队尾指针分别为  $F$  和  $R$ , 其中队头指针  $F$  指向当前队头元素的前一个位置, 队尾指针  $R$  指向当前队尾元素所在的位置, 则出队列的语句为  $F = \underline{\hspace{2cm}}$ ;
  2. 设线性表中有  $n$  个数据元素, 则在顺序存储结构上实现顺序查找的平均时间复杂度为  $\underline{\hspace{2cm}}$ , 在链式存储结构上实现顺序查找的平均时间复杂度为  $\underline{\hspace{2cm}}$ 。
  3. 设一棵二叉树中有  $n$  个结点, 则当用二叉链表作为其存储结构时, 该二叉链表中共有  $\underline{\hspace{2cm}}$  个指针域,  $\underline{\hspace{2cm}}$  个空指针域。

4. 4. 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 B, 则在结点 A 的后面插入结点 B 的操作序列为\_\_\_\_\_。
5. 5. 设无向图 G 中有 n 个顶点和 e 条边, 则其对应的邻接表中有\_\_\_\_\_个表头结点和\_\_\_\_\_个表结点。
6. 6. 设无向图 G 中有 n 个顶点 e 条边, 所有顶点的度数之和为 m, 则 e 和 m 有\_\_\_\_\_关系。
7. 7. 设一棵二叉树的前序遍历序列和中序遍历序列均为 ABC, 则该二叉树的后序遍历序列为\_\_\_\_\_。
8. 8. 设一棵完全二叉树中有 21 个结点, 如果按照从上到下、从左到右的顺序从 1 开始顺序编号, 则编号为 8 的双亲结点的编号是\_\_\_\_\_, 编号为 8 的左孩子结点的编号是\_\_\_\_\_。
9. 9. 下列程序段的功能实现子串 t 在主串 s 中位置的算法, 要求在下划线处填上正确语句。  

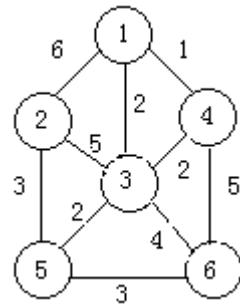
```
int index(char s[ ], char t[ ])
{
    i=j=0;
    while(i<strlen(s) && j<strlen(t)) if(s[i]==t[j]) {i=i+1; j=j+1;} else {i=_____; j=_____;}
    if (j==strlen(t)) return(i-strlen(t)); else return (-1);
}
```
10. 10. 设一个连通图 G 中有 n 个顶点 e 条边, 则其最小生成树上有\_\_\_\_\_条边。

### 三、应用题 (30 分)

1. 设完全二叉树的顺序存储结构中存储数据 ABCDE, 要求给出该二叉树的链式存储结构并给出该二叉树的前序、中序和后序遍历序列。
2. 设给定一个权值集合  $W=(3, 5, 7, 9, 11)$ , 要求根据给定的权值集合构造一棵哈夫曼树并计算哈夫曼树的带权路径长度 WPL。
3. 设一组初始记录关键字序列为(19, 21, 16, 5, 18, 23), 要求给出以 19 为基准的一趟快速排序结果以及第 2 趟直接选择排序后的结果。
4. 设一组初始记录关键字集合为(25, 10, 8, 27, 32, 68), 散列表的长度为 8, 散列函数  $H(k)=k \bmod 7$ , 要求分别用线性探测和链地址法作为解决冲突的方法设计哈希表。
5. 设无向图 G (所右图所示), 要求给出该图的深度优先和广度优先遍历的序列并给出该图的最小生成树。

### 四、算法设计题(20 分)

1. 1. 设计判断单链表中结点是否关于中心对称算法。  
 2. 2. 设计在链式存储结构上建立一棵二叉树的算法。  
 3. 3. 设计判断一棵二叉树是否是二叉排序树的算法。



## 数据结构试卷参考答案

### 一、选择题

- |     |     |     |      |
|-----|-----|-----|------|
| 1.C | 2.C | 3.D | 4.C  |
| 6.C | 7.C | 8.B | 9.B  |
|     |     |     | 10.B |

### 二、填空题

1. 1.  $(F+1) \% m$
2. 2.  $O(n), O(n)$
3. 3.  $2n, n+1$
4. 4.  $s->next=p->next; s->next=s$
5. 5.  $n, 2e$
6. 6.  $m=2e$
7. 7. CBA
8. 8. 4, 16
9. 9.  $i-j+1, 0$
10. 10.  $n-1$

### 三、应用题

1. 1. 链式存储结构略, 前序 ABDEC, 中序 DBEAC, 后序 DEBCA。
2. 2. 哈夫曼树略, WPL=78
3. 3.  $(18, 5, 16, 19, 21, 23), (5, 16, 21, 19, 18, 23)$

	$h_0$
0	$h_1 \rightarrow 8$
1	$h_2$
2	$h_3 \rightarrow 10$
3	$h_4 \rightarrow 25 \rightarrow 32$
4	$h_5 \rightarrow 68$
5	$h_6 \rightarrow 27$
6	链地址法:
7	$\Lambda \quad 8 \quad \Lambda \quad 10 \quad 25 \quad 32 \quad 27 \quad 68$

4. 4. 线性探测:  $\Lambda \quad 8 \quad \Lambda \quad 10 \quad 25 \quad 32 \quad 27 \quad 68$  链地址法:  $h_6 \rightarrow 27$
5. 5. 深度: 125364, 广度: 123456, 最小生成树 T 的边集为  $E=\{(1, 4), (1, 3), (3, 5), (5, 6), (5, 6)\}$

### 四、算法设计题

1. 1. 设计判断单链表中结点是否关于中心对称算法。

```

typedef struct {int s[100]; int top;} sqstack;
int lklistsymmetry(lklist *head)
{
    sqstack stack; stack.top=-1; lklist *p;
    for(p=head;p!=0;p=p->next) {stack.top++; stack.s[stack.top]=p->data;}
    for(p=head;p!=0;p=p->next) if (p->data==stack.s[stack.top]) stack.top=stack.top-1; else return(0);
    return(1);
}

```

2. 2. 设计在链式存储结构上建立一棵二叉树的算法。

```

typedef char datatype;
typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
void createbitree(bitree *&bt)
{
    char ch; scanf("%c",&ch);
    if(ch=='#') {bt=0; return;}
    bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch;
    createbitree(bt->lchild); createbitree(bt->rchild);
}

```

3. 3. 设计判断一棵二叉树是否是二叉排序树的算法。

```

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);}
}

```

## 数据结构试卷 (二)

### 一、选择题(24分)

1. 下面关于线性表的叙述错误的是( )。
  - (A) 线性表采用顺序存储必须占用一片连续的存储空间
  - (B) 线性表采用链式存储不必占用一片连续的存储空间
  - (C) 线性表采用链式存储便于插入和删除操作的实现
  - (D) 线性表采用顺序存储便于插入和删除操作的实现
2. 设哈夫曼树中的叶子结点总数为  $m$ , 若用二叉链表作为存储结构, 则该哈夫曼树中总共有( )个空指针域。
  - (A)  $2m-1$
  - (B)  $2m$
  - (C)  $2m+1$
  - (D)  $4m$
3. 设顺序循环队列  $Q[0: M-1]$  的头指针和尾指针分别为  $F$  和  $R$ , 头指针  $F$  总是指向队头元素的前一位置, 尾指针  $R$  总是指向队尾元素的当前位置, 则该循环队列中的元素个数为( )。
  - (A)  $R-F$
  - (B)  $F-R$
  - (C)  $(R-F+M)\%M$
  - (D)  $(F-R+M)\%M$
4. 设某棵二叉树的中序遍历序列为 ABCD, 前序遍历序列为 CABD, 则后序遍历该二叉树得到序列为( )。
  - (A) BADC
  - (B) BCDA
  - (C) CDAB
  - (D) CBDA
5. 设某完全无向图中有  $n$  个顶点, 则该完全无向图中有( )条边。
  - (A)  $n(n-1)/2$
  - (B)  $n(n-1)$
  - (C)  $n^2$
  - (D)  $n^2-1$
6. 设某棵二叉树中有 2000 个结点, 则该二叉树的最小高度为( )。
  - (A) 9
  - (B) 10
  - (C) 11
  - (D) 12
7. 设某有向图中有  $n$  个顶点, 则该有向图对应的邻接表中有( )个表头结点。
  - (A)  $n-1$
  - (B)  $n$
  - (C)  $n+1$
  - (D)  $2n-1$
8. 设一组初始记录关键字序列为(5, 2, 6, 3, 8), 以第一个记录关键字 5 为基准进行一趟快速排序的结果为( )。
  - (A) 2, 3, 5, 8, 6
  - (B) 3, 2, 5, 8, 6
  - (C) 3, 2, 5, 6, 8
  - (D) 2, 3, 6, 5, 8

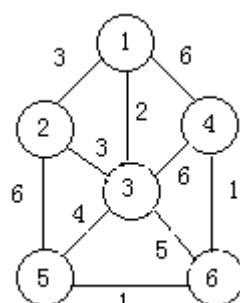
### 二、填空题(24分)

1. 1. 为了能有效地应用 HASH 查找技术, 必须解决的两个问题是\_\_\_\_\_和\_\_\_\_\_。
2. 2. 下面程序段的功能实现数据  $x$  进栈, 要求在下划线处填上正确的语句。

```
typedef struct {int s[100]; int top;} sqstack;
void push(sqstack &stack,int x)
{
    if(stack.top==m-1) printf("overflow");
    else {_____; _____;
}
```
3. 中序遍历二叉排序树所得到的序列是\_\_\_\_\_序列(填有序或无序)。
4. 快速排序的最坏时间复杂度为\_\_\_\_\_, 平均时间复杂度为\_\_\_\_\_.  
5. 设某棵二叉树中度数为 0 的结点数为  $N_0$ , 度数为 1 的结点数为  $N_1$ , 则该二叉树中度数为 2 的结点数为\_\_\_\_\_; 若采用二叉链表作为该二叉树的存储结构, 则该二叉树中共有\_\_\_\_\_个空指针域。
6. 设某无向图中顶点数和边数分别为  $n$  和  $e$ , 所有顶点的度数之和为  $d$ , 则  $e=$ \_\_\_\_\_。
7. 设一组初始记录关键字序列为(55, 63, 44, 38, 75, 80, 31, 56), 则利用筛选法建立的初始堆为\_\_\_\_\_。
8. 8. 设某无向图  $G$  的邻接表为  $v_4 -> 1 -> 3$ , 则从顶点  $V_1$  开始的深度优先遍历序列为\_\_\_\_\_; 广度优先遍历序列为\_\_\_\_\_。

### 三、应用题(36分)

1. 1. 设一组初始记录关键字序列为(45, 80, 48, 40, 22, 78), 则分别给出第 4 趟简单选择排序和第 4 趟直接插入排序后的结果。
2. 2. 设指针变量  $p$  指向双向链表中结点 A, 指针变量  $q$  指向被插入结点 B, 要求给出在结点 A 的后面插入结点 B 的操作序列(设双向链表中结点的两个指针域分别为 llink 和 rlink)。
3. 3. 设一组有序的记录关键字序列为(13, 18, 24, 35, 47, 50, 62, 83, 90), 查找方法用二分查找, 要求计算出查找关键字 62 时的比较次数并计算出查找成功时的平均查找长度。
4. 4. 设一棵树  $T$  中边的集合为  $\{(A, B), (A, C), (A, D), (B, E), (C, F), (C, G)\}$ , 要求用孩子兄弟表示法(二叉链表)表示出该树的存储结构并将该树转化成对应的二叉树。
5. 5. 设有无向图  $G$  (如右图所示), 要求给出用普里姆算法构造最小生成树所走过的边的集合。



6 . 6 . 设有一组初始记录关键字为(45, 80, 48, 40, 22, 78), 要求构造一棵二叉排序树并给出构造过程。

**四、算法设计题(16分)**

- 1 . 1 . 设有一组初始记录关键字序列  $(K_1, K_2, \dots, K_n)$  , 要求设计一个算法能够在  $O(n)$  的时间复杂度内将线性表划分成两部分, 其中左半部分的每个关键字均小于  $K_i$ , 右半部分的每个关键字均大于等于  $K_i$ 。
- 2 . 2 . 设有两个集合 A 和集合 B, 要求设计生成集合  $C=A\cap B$  的算法, 其中集合 A、B 和 C 用链式存储结构表示。

## 数据结构试卷（二）参考答案

### 一、选择题

1.D      2.B      3.C      4.A      5.A      6.C      7.B      8.C

### 二、填空题

1. 1. 构造一个好的 HASH 函数，确定解决冲突的方法
2. 2. stack.top++, stack.s[stack.top]=x
3. 3. 有序
4. 4.  $O(n^2)$ ,  $O(n \log_2 n)$
5. 5.  $N_0-1$ ,  $2N_0+N_1$
6. 6.  $d/2$
7. 7. (31, 38, 54, 56, 75, 80, 55, 63)
8. 8. (1, 3, 4, 2), (1, 3, 2, 4)

### 三、应用题

1. 1. (22, 40, 45, 48, 80, 78), (40, 45, 48, 80, 22, 78)
2. 2.  $q->llink=p; q->rlink=p->rlink; p->rlink->llink=q; p->rlink=q;$
3. 3.  $2, ASL=91*1+2*2+3*4+4*2=25/9$
4. 4. 树的链式存储结构略，二叉树略
5. 5.  $E=\{(1, 3), (1, 2), (3, 5), (5, 6), (6, 4)\}$
6. 6. 略

### 四、算法设计题

1. 1. 设有一组初始记录关键字序列  $(K_1, K_2, \dots, K_n)$ ，要求设计一个算法能够在  $O(n)$  的时间复杂度内将线性表划分成两部分，其中左半部分的每个关键字均小于  $K_i$ ，右半部分的每个关键字均大于等于  $K_i$ 。

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

2. 2. 设有两个集合 A 和集合 B，要求设计生成集合  $C=A \cap B$  的算法，其中集合 A、B 和 C 用链式存储结构表示。

```
typedef struct node {int data; struct node *next;}lklist;
void intersection(lklist *ha,lklist *hb,lklist *&hc)
{
    lklist *p,*q,*t;
    for(p=ha, hc=0;p!=0;p=p->next)
    {
        for(q=hb;q!=0;q=q->next) if (q->data==p->data) break;
        if(q!=0){ t=(lklist *)malloc(sizeof(lklist)); t->data=p->data;t->next=hc; hc=t;}
    }
}
```

### 数据结构试卷 (三)

#### 一、选择题(30 分)

1. 设某数据结构的二元组形式表示为  $A=(D, R)$ ,  $D=\{01, 02, 03, 04, 05, 06, 07, 08, 09\}$ ,  $R=\{r\}$ ,  $r=\{\langle 01, 02 \rangle, \langle 01, 03 \rangle, \langle 01, 04 \rangle, \langle 02, 05 \rangle, \langle 02, 06 \rangle, \langle 03, 07 \rangle, \langle 03, 08 \rangle, \langle 03, 09 \rangle\}$ , 则数据结构 A 是 ( )。  
 (A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构
2. 下面程序的时间复杂度为 ( )  
 for ( $i=1, s=0; i \leq n; i++$ ) { $t=1;$  for ( $j=1; j \leq i; j++$ )  $t=t*j;$   $s=s+t;$  }  
 (A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n^3)$  (D)  $O(n^4)$
3. 设指针变量 p 指向单链表中结点 A, 若删除单链表中结点 A, 则需要修改指针的操作序列为 ( )。  
 (A)  $q=p->next; p->data=q->data; p->next=q->next; free(q);$   
 (B)  $q=p->next; q->data=p->data; p->next=q->next; free(q);$   
 (C)  $q=p->next; p->next=q->next; free(q);$   
 (D)  $q=p->next; p->data=q->data; free(q);$
4. 设有 n 个待排序的记录关键字, 则在堆排序中需要 ( ) 个辅助记录单元。  
 (A) 1 (B) n (C)  $n \log_2 n$  (D)  $n^2$
5. 设一组初始关键字记录关键字为 (20, 15, 14, 18, 21, 36, 40, 10), 则以 20 为基准记录的一趟快速排序结束后的结果为 ( )。  
 (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
6. 设二叉排序树中有 n 个结点, 则在二叉排序树的平均平均查找长度为 ( )。  
 (A)  $O(1)$  (B)  $O(\log_2 n)$  (C) (D)  $O(n^2)$
7. 设无向图 G 中有 n 个顶点 e 条边, 则其对应的邻接表中的表头结点和表结点的个数分别为 ( )。  
 (A) n, e (B) e, n (C)  $2n, e$  (D) n,  $2e$
8. 设某强连通图中有 n 个顶点, 则该强连通图中至少有 ( ) 条边。  
 (A)  $n(n-1)$  (B)  $n+1$  (C) n (D)  $n(n+1)$
9. 设有 5000 个待排序的记录关键字, 如果需要用最快的方法选出其中最小的 10 个记录关键字, 则用下列 ( ) 方法可以达到此目的。  
 (A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 插入排序
10. 下列四种排序中 ( ) 的空间复杂度最大。  
 (A) 插入排序 (B) 冒泡排序 (C) 堆排序 (D) 归并排序

#### 二、填空题(48 分, 其中最后两小题各 6 分)

1. 1. 数据的物理结构主要包括 和 两种情况。
2. 2. 设一棵完全二叉树中有 500 个结点, 则该二叉树的深度为 ; 若用二叉链表作为该完全二叉树的存储结构, 则共有 个空指针域。
3. 3. 设输入序列为 1、2、3, 则经过栈的作用后可以得到 种不同的输出序列。
4. 4. 设有向图 G 用邻接矩阵  $A[n][n]$  作为存储结构, 则该邻接矩阵中第 i 行上所有元素之和等于顶点 i 的 , 第 i 列上所有元素之和等于顶点 i 的 。
5. 5. 设哈夫曼树中共有 n 个结点, 则该哈夫曼树中有 个度数为 1 的结点。
6. 6. 设有向图 G 中有 n 个顶点 e 条有向边, 所有的顶点入度数之和为 d, 则 e 和 d 的关系为 。
7. 7. 遍历二叉排序树中的结点可以得到一个递增的关键字序列 (填先序、中序或后序)。
8. 8. 设查找表中有 100 个元素, 如果用二分法查找方法查找数据元素 X, 则最多需要比较 次就可以断定数据元素 X 是否在查找表中。
9. 9. 不论是顺序存储结构的栈还是链式存储结构的栈, 其入栈和出栈操作的时间复杂度均为 。
10. 10. 设有 n 个结点的完全二叉树, 如果按照从上到下、从左到右从 1 开始顺序编号, 则第 i 个结点的双亲结点编号为 , 右孩子结点的编号为 。
11. 11. 设一组初始记录关键字为 (72, 73, 71, 23, 94, 16, 5), 则以记录关键字 72 为基准的一趟快速排序结果为 。
12. 12. 设有向图 G 中有向边的集合  $E=\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 4 \rangle, \langle 4, 2 \rangle, \langle 4, 3 \rangle\}$ , 则该图的一种拓扑序列为 。
13. 13. 下列算法实现在顺序散列表中查找值为 x 的关键字, 请在下划线处填上正确的语句。
- ```
struct record{int key; int others;};
int hashsqsearch(struct record hashtable[ ], int k)
{
    int i,j; j=i=k % p;
    while (hashtable[j].key!=k&&hashtable[j].flag!=0){j=(_____) % m; if (i==j) return(-1);}
    if (_____ ) return(j); else return(-1);
}
```
14. 14. 下列算法实现在二叉排序树上查找关键值 k, 请在下划线处填上正确的语句。

```

typedef struct node{int key; struct node *lchild; struct node *rchild;}bitree;
bitree *bstsearch(bitree *t, int k)
{
    if(t==0) return(0);else while (t!=0)
        if (t->key==k) _____; else if (t->key>k) t=t->lchild; else _____;
}

```

### 三、算法设计题(22分)

1. 1. 设计在单链表中删除值相同的多余结点的算法。
2. 2. 设计一个求结点 x 在二叉树中的双亲结点算法。

### 数据结构试卷 (三) 参考答案

#### 一、选择题

|     |     |     |     |      |
|-----|-----|-----|-----|------|
| 1.B | 2.B | 3.A | 4.A | 5.A  |
| 6.B | 7.D | 8.C | 9.B | 10.D |

第3小题分析：首先用指针变量 q 指向结点 A 的后继结点 B，然后将结点 B 的值复制到结点 A 中，最后删除结点 B。

第9小题分析：9 快速排序、归并排序和插入排序必须等到整个排序结束后才能够求出最小的 10 个数，而堆排序只需要在初始堆的基础上再进行 10 次筛选即可，每次筛选的时间复杂度为  $O(\log_2 n)$ 。

#### 二、填空题

1. 1. 顺序存储结构、链式存储结构
2. 2. 9, 501
3. 3. 5
4. 4. 出度, 入度
5. 5. 0
6. 6. e=d
7. 7. 中序
8. 8. 7
9. 9.  $O(1)$
10. 10.  $i/2$ ,  $2i+1$
11. 11. (5, 16, 71, 23, 72, 94, 73)
12. 12. (1, 4, 3, 2)
13. 13.  $j+1$ , hashtable[j].key==k
14. 14. return(t), t=t->rchild

第8小题分析：二分查找的过程可以用一棵二叉树来描述，该二叉树称为二叉判定树。在有序表上进行二分查找时的查找长度不超过二叉判定树的高度  $1+\log_2 n$ 。

#### 三、算法设计题

1. 1. 设计在单链表中删除值相同的多余结点的算法。

```

typedef int datatype;
typedef struct node {datatype data; struct node *next;}lklist;
void delredundant(lklist *&head)
{
    lklist *p,*q,*s;
    for(p=head;p!=0;p=p->next)
    {
        for(q=p->next,s=q;q!=0; )
        if (q->data==p->data) {s->next=q->next; free(q);q=s->next;}
        else {s=q,q=q->next;}
    }
}

```

2. 2. 设计一个求结点 x 在二叉树中的双亲结点算法。

```

typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
bitree *q[20]; int r=0,f=0,flag=0;
void preorder(bitree *bt, char x)
{
    if (bt!=0 && flag==0)
        if (bt->data==x) { flag=1; return; }
        else {r=(r+1)% 20; q[r]=bt; preorder(bt->lchild,x); preorder(bt->rchild,x); }
}
void parent(bitree *bt,char x)
{
    int i;
    preorder(bt,x);
}

```

```

for(i=f+1; i<=r; i++) if ((q[i]->lchild->data==x || q[i]->rchild->data) break;
if (flag==0) printf("not found x\n");
else if (i<=r) printf("%c",bt->data); else printf("not parent");
}

```

数据结构试卷 (四)

### 一、选择题(30分)



## 二、填空题(42分)

1. 设有 n 个无序的记录关键字，则直接插入排序的时间复杂度为\_\_\_\_\_，快速排序的平均时间复杂度为\_\_\_\_\_。
  2. 设指针变量 p 指向双向循环链表中的结点 X，则删除结点 X 需要执行的语句序列为\_\_\_\_\_ (设结点中的两个指针域分别为 llink 和 rlink)。
  3. 根据初始关键字序列(19, 22, 01, 38, 10)建立的二叉排序树的高度为\_\_\_\_\_。
  4. 深度为 k 的完全二叉树中最少有\_\_\_\_\_个结点。
  5. 设初始记录关键字序列为(K<sub>1</sub>, K<sub>2</sub>, …, K<sub>n</sub>)，则用筛选法思想建堆必须从第\_\_\_\_\_个元素开始进行筛选。
  6. 设哈夫曼树中共有 99 个结点，则该树中有\_\_\_\_\_个叶子结点；若采用二叉链表作为存储结构，则该树中有\_\_\_\_\_个空指针域。
  7. 设有一个顺序循环队列中有 M 个存储单元，则该循环队列中最多能够存储\_\_\_\_\_个队列元素；当前实际存储\_\_\_\_\_个队列元素（设头指针 F 指向当前队头元素的前一个位置，尾指针指向当前队尾元素的位置）。
  8. 设顺序线性表中有 n 个数据元素，则第 i 个位置上插入一个数据元素需要移动表中\_\_\_\_\_个数据元素；删除第 i 个位置上的数据元素需要移动表中\_\_\_\_\_个元素。
  9. 设一组初始记录关键字序列为(20, 18, 22, 16, 30, 19)，则以 20 为中轴的一趟快速排序结果为\_\_\_\_\_。
  10. 设一组初始记录关键字序列为(20, 18, 22, 16, 30, 19)，则根据这些初始关键字序列建成的初始堆为\_\_\_\_\_。
  11. 设某无向图 G 中有 n 个顶点，用邻接矩阵 A 作为该图的存储结构，则顶点 i 和顶点 j 互为邻接点的条件是\_\_\_\_\_。
  12. 设无向图对应的邻接矩阵为 A，则 A 中第 i 行上非 0 元素的个数\_\_\_\_\_第 i 列上非 0 元素的个数（填等于、大于或小于）。
  13. 设前序遍历某二叉树的序列为 ABCD，中序遍历该二叉树的序列为 BADC，则后序遍历该二叉树的序列为\_\_\_\_\_。
  14. 设散列函数 H(k)=k mod p，解决冲突的方法为链地址法。要求在下列算法划线处填上正确的语句完成在散列表 hashtable 中查找关键字值等于 k 的结点，成功时返回指向关键字的指针，不成功时返回标志 0。  
typedef struct node {int key; struct node \*next;} lklist;  
void createlkhash(lklist \*hashtable[ ]) {  
 int i,k; lklist \*s;

```

for(i=0;i<m;i++)_____;
for(i=0;i<n;i++)
{
    s=(lklist *)malloc(sizeof(lklist)); s->key=a[i];
    k=a[i] % p; s->next=hashtable[k];_____;
}
}

```

### 三、算法设计题(28分)

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。
2. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。
3. 在链式存储结构上建立一棵二叉排序树。

### 数据结构试卷 (四) 参考答案

#### 一、选择题

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| 1 . C | 2 . D | 3 . D | 4 . B | 5 . C  |
| 6 . A | 7 . B | 8 . A | 9 . C | 10 . A |

#### 二、填空题

1.  $O(n^2)$ ,  $O(n \log_2 n)$
2.  $p->llink->rlink=p->rlink; p->rlink->llink=p->llink$
3. 3
4.  $2^{k-1}$
5.  $n/2$
6. 50, 51
7.  $m-1$ ,  $(R-F+M)\%M$
8.  $n+1-i$ ,  $n-i$
9. (19, 18, 16, 20, 30, 22)
10. (16, 18, 19, 20, 32, 22)
11.  $A[i][j]=1$
12. 等于
13. BDCA
14.  $hashtable[i]=0$ ,  $hashtable[k]=s$

#### 三、算法设计题

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。

```

typedef char datatype;
typedef struct node {datatype data; struct node *next;} lklist;
void split(lklist *head,lklist *&ha,lklist *&hb,lklist *&hc)
{
    lklist *p; ha=0,hb=0,hc=0;
    for(p=head;p!=0;p=head)
    {
        head=p->next; p->next=0;
        if (p->data>='A' && p->data<='Z') {p->next=ha; ha=p;}
        else if (p->data>='0' && p->data<='9') {p->next=hb; hb=p;} else {p->next=hc; hc=p;}
    }
}

```

2. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。

```

typedef struct node {int data; struct node *lchild,*rchild;} bitree;
void swapbitree(bitree *bt)
{
    bitree *p;
    if(bt==0) return;
    swapbitree(bt->lchild); swapbitree(bt->rchild);
    p=bt->lchild; bt->lchild=bt->rchild; bt->rchild=p;
}

```

3. 在链式存储结构上建立一棵二叉排序树。

```

#define n 10
typedef struct node{int key; struct node *lchild,*rchild;} bitree;
void bstinsert(bitree *&bt,int key)
{
    if (bt==0){bt=(bitree *)malloc(sizeof(bitree)); bt->key=key;bt->lchild=bt->rchild=0;}
    else if (bt->key>key) bstinsert(bt->lchild,key); else bstinsert(bt->rchild,key);
}

```

```

}
void createbstree(bitree *&bt)
{
    int i;
    for(i=1;i<=n;i++) bstinsert(bt,random(100));
}

```

### 数据结构试卷 (五)

#### 一、选择题(30分)

1. 数据的最小单位是( )。
 

(A) 数据项      (B) 数据类型      (C) 数据元素      (D) 数据变量
2. 设一组初始记录关键字序列为(50, 40, 95, 20, 15, 70, 60, 45), 则以增量d=4的一趟希尔排序结束后前4条记录关键字为( )。
 

(A) 40, 50, 20, 95      (B) 15, 40, 60, 20  
  (C) 15, 20, 40, 45      (D) 45, 40, 15, 20
3. 设一组初始记录关键字序列为(25, 50, 15, 35, 80, 85, 20, 40, 36, 70), 其中含有5个长度为2的有序子表, 则用归并排序的方法对该记录关键字序列进行一趟归并后的结果为( )。
 

(A) 15, 25, 35, 50, 20, 40, 80, 85, 36, 70  
  (B) 15, 25, 35, 50, 80, 20, 85, 40, 70, 36  
  (C) 15, 25, 35, 50, 80, 85, 20, 36, 40, 70  
  (D) 15, 25, 35, 50, 80, 20, 36, 40, 70, 85
4. 函数substr("DATASTRUCTURE", 5, 9)的返回值为( )。
 

(A) "STRUCTURE"      (B) "DATA"  
  (C) "ASTRUCTUR"      (D) "DATASTRUCTURE"
5. 设一个有序的单链表中有n个结点, 现要求插入一个新结点后使得单链表仍然保持有序, 则该操作的时间复杂度为( )。
 

(A) O(log<sub>2</sub>n)      (B) O(1)      (C) O(n<sup>2</sup>)      (D) O(n)
6. 设一棵m叉树中度数为0的结点数为N<sub>0</sub>, 度数为1的结点数为N<sub>1</sub>, ……, 度数为m的结点数为N<sub>m</sub>, 则N<sub>0</sub>= ( )。
 

(A) N<sub>1</sub>+N<sub>2</sub>+……+N<sub>m</sub>      (B) 1+N<sub>2</sub>+2N<sub>3</sub>+3N<sub>4</sub>+……+(m-1)N<sub>m</sub>  
  (C) N<sub>2</sub>+2N<sub>3</sub>+3N<sub>4</sub>+……+(m-1)N<sub>m</sub>      (D) 2N<sub>1</sub>+3N<sub>2</sub>+……+(m+1)N<sub>m</sub>
7. 设有序表中有1000个元素, 则用二分查找查找元素X最多需要比较( )次。
 

(A) 25      (B) 10      (C) 7      (D) 1
8. 设连通图G中的边集E={(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)}, 则从顶点a出发可以得到一种深度优先遍历的顶点序列为( )。
 

(A) abedfc      (B) acfebd      (C) aebdfc      (D) aedfcb
9. 设输入序列是1、2、3、……、n, 经过栈的作用后输出序列的第一个元素是n, 则输出序列中第i个输出元素是( )。
 

(A) n-i      (B) n-1-i      (C) n+1-i      (D) 不能确定
10. 设一组初始记录关键字序列为(45, 80, 55, 40, 42, 85), 则以第一个记录关键字45为基准而得到一趟快速排序的结果是( )。
 

(A) 40, 42, 45, 55, 80, 83      (B) 42, 40, 45, 80, 85, 88  
  (C) 42, 40, 45, 55, 80, 85      (D) 42, 40, 45, 85, 55, 80

#### 二、填空题(共30分)

1. 设有一个顺序共享栈S[0: n-1], 其中第一个栈顶指针top1的初值为-1, 第二个栈顶指针top2的初值为n, 则判断共享栈满的条件是\_\_\_\_\_。
2. 在图的邻接表中用顺序存储结构存储表头结点的优点是\_\_\_\_\_。
3. 设有一个n阶的下三角矩阵A, 如果按照行的顺序将下三角矩阵中的元素(包括对角线上元素)存放在n(n+1)个连续的存储单元中, 则A[i][j]与A[0][0]之间有\_\_\_\_\_个数据元素。
4. 栈的插入和删除只能在栈的栈顶进行, 后进栈的元素必定先出栈, 所以又把栈称为\_\_\_\_\_表; 队列的插入和删除运算分别在队列的两端进行, 先进队列的元素必定先出队列, 所以又把队列称为\_\_\_\_\_表。
5. 设一棵完全二叉树的顺序存储结构中存储数据元素为ABCDEF, 则该二叉树的前序遍历序列为\_\_\_\_\_, 中序遍历序列为\_\_\_\_\_, 后序遍历序列为\_\_\_\_\_。
6. 设一棵完全二叉树有128个结点, 则该完全二叉树的深度为\_\_\_\_\_, 有\_\_\_\_\_个叶子结点。
7. 设有向图G的存储结构用邻接矩阵A来表示, 则A中第i行中所有非零元素个数之和等于顶点i的\_\_\_\_\_, 第i列中所有非零元素个数之和等于顶点i的\_\_\_\_\_。
8. 设一组初始记录关键字序列(k<sub>1</sub>, k<sub>2</sub>, ……, k<sub>n</sub>)是堆, 则对i=1, 2, …, n/2而言满足的条件为\_\_\_\_\_。
9. 下面程序段的功能是实现冒泡排序算法, 请在下划线处填上正确的语句。
 

```

void bubble(int r[n])
{
    for(i=1;i<=n-1; i++)

```

```

    {
        for(exchange=0,j=0; j<_____ ;j++)
            if (r[j]>r[j+1]) {temp=r[j+1]; _____ ;r[j]=temp;exchange=1;}
        if (exchange==0) return;
    }
}

```

10. 下面程序段的功能是实现二分查找算法, 请在下划线处填上正确的语句。

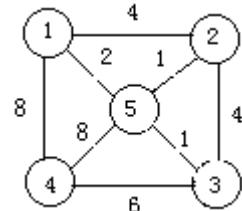
```

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

```

### 三、应用题(24分)

1. 设某棵二叉树的中序遍历序列为 DBEAC, 前序遍历序列为 ABDEC, 要求给出该二叉树的后序遍历序列。
2. 设无向图 G (如右图所示), 给出该图的最小生成树上边的集合并计算最小生成树各边上的权值之和。
3. 设一组初始记录关键字序列为(15, 17, 18, 22, 35, 51, 60), 要求计算出成功查找时的平均查找长度。
4. 设散列表的长度为 8, 散列函数  $H(k)=k \bmod 7$ , 初始记录关键字序列为(25, 31, 8, 27, 13, 68), 要求分别计算用线性探测法和链地址法作为解决冲突方法的平均查找长度。



### 四、算法设计题(16分)

1. 设计判断两个二叉树是否相同的算法。
2. 设计两个有序单链表的合并排序算法。

## 数据结构试卷 (五) 参考答案

### 一、选择题

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| 1 . A | 2 . B | 3 . A | 4 . A | 5 . D  |
| 6 . B | 7 . B | 8 . B | 9 . C | 10 . C |

### 二、填空题

1.  $\text{top1}+1=\text{top2}$
2. 可以随机访问到任一个顶点的简单链表
3.  $i(i+1)/2+j-1$
4. FILO, FIFO
5. ABDECF, DBEAFC, DEBFCA
6. 8, 64
7. 出度, 入度
8.  $k_i \leq k_{2i} \&\& k_i \leq k_{2i+1}$
9.  $n-i, r[j+1]=r[j]$
10.  $\text{mid}=(\text{low}+\text{high})/2, r[\text{mid}].\text{key}>k$

### 三、应用题

1. DEBCA
2.  $E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$
3.  $ASL=(1*1+2*2+3*4)/7=17/7$
4.  $ASL1=7/6, ASL2=4/3$

### 四、算法设计题

1. 设计判断两个二叉树是否相同的算法。
 

```

typedef struct node {
datatype data;
struct node *lchild,*rchild;} bitree;
int judgebitree(bitree *bt1,bitree *bt2)
{
    if (bt1==0 && bt2==0) return(1);
    else if (bt1==0 || bt2==0 || bt1->data!=bt2->data) return(0);
    else return(judgebitree(bt1->lchild,bt2->lchild)*judgebitree(bt1->rchild,bt2->rchild));
}

```
2. 设计两个有序单链表的合并排序算法。

```
void mergelklist(lklist *ha,lklist *hb,lklist *&hc)
{
    lklist *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;
}
```

数据结构试卷 (六)

### 一、选择题(30分)

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 ( )。  
 (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) [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)  $head==0$  (B)  $head->next==0$   
 (C)  $head->next==head$  (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) 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 ( )。  
 (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 ( )。  
 (A) 堆排序 (B) 冒泡排序 (C) 快速排序 (D) 希尔排序

7. 设某棵三叉树中有 40 个结点, 则该三叉树的最小高度为 ( )。  
 (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)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n^{1/2})$  (D)  $O(1 \log_2 n)$

9. 二路归并排序的时间复杂度为 ( )。  
 (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 ( ) 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 ( )。  
 (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)  $O(n+e)$  (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 ( ) 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 ( )。  
 (A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n \log_2 n)$  (D)  $O(1 \log_2 n)$

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 ( )。  
 (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(20分)

1. 调用一次深度优先遍历可以访问到图中的所有顶点。 ( )
  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。 ( )
  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。 ( )
  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。 ( )
  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。 ( )
  6. 层次遍历初始堆可以得到一个有序的序列。 ( )
  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。 ( )
  8. 线性表的顺序存储结构比链式存储结构更好。 ( )
  9. 中序遍历二叉排序树可以得到一个有序的序列。 ( )
  10. 快速排序是排序算法中平均性能最好的一种排序。 ( )

### 三、填空题(30分)

1. for( $i=1$ ,  $t=1$ ,  $s=0$ ;  $i \leq n$ ;  $i++$ ) { $t=t*i$ ;  $s=s+t$ ; } 的时间复杂度为\_\_\_\_\_。

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\_\_\_\_\_ (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为\_\_\_\_\_。

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\_\_\_\_\_。

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\_\_\_\_\_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\_\_\_\_\_。

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\_\_\_\_\_。
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\_\_\_\_\_。
9. 快速排序算法的空间复杂度平均情况下为\_\_\_\_\_，最坏的情况下为\_\_\_\_\_。
10. 散列表中解决冲突的两种方法是\_\_\_\_\_和\_\_\_\_\_。

#### 四、算法设计题(20分)

1. 1. 设计在顺序有序表中实现二分查找的算法。
2. 2. 设计判断二叉树是否为二叉排序树的算法。
3. 3. 在链式存储结构上设计直接插入排序算法

#### 数据结构试卷（六）参考答案

##### 一、选择题

1. D      2. A      3. A      4. A      5. D  
 6. D      7. B      8. A      9. C      10. B  
 11. C 12. A      13. B      14. D      15. B

##### 二、判断题

1. 错 2. 对 3. 对 4. 对 5. 错  
 6. 错 7. 对 8. 错 9. 对 10. 对

##### 三、填空题

1. 1.  $O(n)$   
 2. 2.  $s->next=p->next; p->next=s$   
 3. 3. (1, 3, 2, 4, 5)  
 4. 4.  $n-1$   
 5. 5. 129  
 6. 6.  $F=R$   
 7. 7.  $p->lchild==0 \&\& p->rchild==0$   
 8. 8.  $O(n^2)$   
 9. 9.  $O(n \log_2 n), O(n)$   
 10. 10. 开放定址法，链地址法

#### 四、算法设计题

1. 1. 设计在顺序有序表中实现二分查找的算法。
- ```
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. 2. 设计判断二叉树是否为二叉排序树的算法。
- ```
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. 3. 在链式存储结构上设计直接插入排序算法
- ```
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;}
    }
}
```

## 数据结构试卷 (七)

### 一、选择题(30分)

1. 设某无向图有  $n$  个顶点，则该无向图的邻接表中有（ ）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（ ）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（ ）。  
(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. （ ）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（ ）。  
(A)  $2i+1$       (B)  $2i$       (C)  $i/2$       (D)  $2i-1$
6. 程序段  $s=i=0; do \{i=i+1; s=s+i; \}while(i<=n);$  的时间复杂度为（ ）。  
(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（ ）。  
(A)  $head==0$       (B)  $head->next==0$   
(C)  $head->next==head$       (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（ ）。  
(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（ ）。  
(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（ ）。  
(A)  $top=top+1;$       (B)  $top=top-1;$   
(C)  $top->next=top;$       (D)  $top=top->next;$

### 二、判断题(20分)

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。（ ）
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。（ ）
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  $O(\log_2 n)$ 。（ ）
4. 完全二叉树中的叶子结点只可能在最后两层中出现。（ ）
5. 哈夫曼树中没有度数为 1 的结点。（ ）
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。（ ）
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。（ ）
8. 由树转化成二叉树，该二叉树的右子树不一定为空。（ ）
9. 线性表中的所有元素都有一个前驱元素和后继元素。（ ）
10. 带权无向图的最小生成树是唯一的。（ ）

### 三、填空题(30分)

1. 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 \_\_\_\_\_ =  $p$ ;  $s->right=p->right$ ; \_\_\_\_\_ =  $s$ ;  $p->right->left=s$ ；  
(设结点中的两个指针域分别为  $left$  和  $right$ )。
2. 2. 设完全有向图中有  $n$  个顶点，则该完全有向图中共有 \_\_\_\_\_ 条有向弧；设完全无向图中有  $n$  个顶点，则该完全无向图中共有 \_\_\_\_\_ 条无向边。
3. 3. 设关键字序列为( $K_1, K_2, \dots, K_n$ )，则用筛选法建初始堆必须从第 \_\_\_\_\_ 个元素开始进行筛选。
4. 4. 解决散列表冲突的两种方法是 \_\_\_\_\_ 和 \_\_\_\_\_。
5. 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 \_\_\_\_\_ 个。
6. 6. 高度为  $h$  的完全二叉树中最少有 \_\_\_\_\_ 个结点，最多有 \_\_\_\_\_ 个结点。
7. 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是 \_\_\_\_\_。
8. 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是 \_\_\_\_\_。
9. 9. 设一棵二叉树的前序序列为 ABC，则有 \_\_\_\_\_ 种不同的二叉树可以得到这种序列。
10. 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=i+1; if (i<j) {r[j]=r[i];j=j-1;}
}
_____;
}

```

#### 四、算法设计题(20分)

1. 1. 设计在链式结构上实现简单选择排序算法。
2. 2. 设计在顺序存储结构上实现求子串算法。
3. 3. 设计求结点在二叉排序树中层次的算法。

### 数据结构试卷 (七)

#### 一、选择题

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| 1 . B | 2 . B | 3 . C | 4 . B | 5 . B  |
| 6 . A | 7 . C | 8 . C | 9 . B | 10 . D |

#### 二、判断题

- |       |       |       |       |        |
|-------|-------|-------|-------|--------|
| 1 . 对 | 2 . 对 | 3 . 对 | 4 . 对 | 5 . 对  |
| 6 . 对 | 7 . 对 | 8 . 错 | 9 . 错 | 10 . 错 |

#### 三、填空题

1. 1. s->left=p, p->right
2. 2. n(n-1), n(n-1)/2
3. 3. n/2
4. 4. 开放定址法, 链地址法
5. 5. 14
6. 6.  $2^{h-1}$ ,  $2^h-1$
7. 7. (12, 24, 35, 27, 18, 26)
8. 8. (12, 18, 24, 27, 35, 26)
9. 9. 5
10. 10. i<j && r[i].key<x.key, r[i]=x

#### 四、算法设计题

1. 1. 设计在链式结构上实现简单选择排序算法。

```

void simpleselectsortlklist(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. 2. 设计在顺序存储结构上实现求子串算法。

```

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. 3. 设计求结点在二叉排序树中层次的算法。

```

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);}
}

```

### 数据结构试卷 (八)

#### 一、选择题(30分)

1. 1. 字符串的长度是指 ( ) 。
 

(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 ( )
 

(A) O(n)	(B) O(1)	(C) O( $n^2$ )	(D) O(log <sub>2</sub> n)
----------	----------	----------------	---------------------------
3. 两个字符串相等的充要条件是 ( ) 。
 

(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 ( ) 。
 

(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 ( ) 。
 

(A) O(n)	(B) O(log <sub>2</sub> n)	(C) O(nlog <sub>2</sub> 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 ( ) 。
 

(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 ( ) 。
 

(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, 则该三叉树中有 ( ) 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) aedfcba	(B) acfbed	(C) aebcfda	(D) aedfbca
-------------	------------	-------------	-------------
10. 队列是一种 ( ) 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(20 分)

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。 ( )
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(nlog<sub>2</sub>n)。 ( )
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。 ( )
4. 二维数组和多维数组均不是特殊的线性结构。 ( )
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。 ( )
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。 ( )
7. 非空的双向循环链表中任何结点的前驱指针均不为空。 ( )
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)。 ( )
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。 ( )
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。 ( )

## 三、填空题(30 分)

1. 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 \_\_\_\_\_。
2. 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->data=k;t->lchild=t->rchild=0;}
    else if(t->data>k) bstinsert(t->lchild,k);else _____;
}
```
3. 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; \_\_\_\_\_。
4. 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=\_\_\_\_\_ 和 head=\_\_\_\_\_ (设结点中的两个指针域分别为 llink 和 rlink)。
5. 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 \_\_\_\_\_。
6. 6. 完全二叉树中第 5 层上最少有 \_\_\_\_\_ 个结点, 最多有 \_\_\_\_\_ 个结点。
7. 7. 设有向图中不存在有向边<V<sub>i</sub>, V<sub>j</sub>>, 则其对应的邻接矩阵 A 中的数组元素 A[i][j] 的值等于 \_\_\_\_\_。
8. 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 \_\_\_\_\_。
9. 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 \_\_\_\_\_ 条边。

10. 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\_\_\_\_\_相互交换即可。

#### 四、算法设计题(20分)

1. 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。
2. 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。

### 数据结构试卷 (八) 参考答案

#### 一、选择题

1. C 2. C 3. C 4. B 5. B
6. C 7. B 8. C 9. A 10. A

#### 二、判断题

1. 对 2. 错 3. 对 4. 错 5. 错
6. 对 7. 对 8. 对 9. 对 10. 对

#### 三、填空题

1. 1. (49, 13, 27, 50, 76, 38, 65, 97)
2. 2. t=(bitree \*)malloc(sizeof(bitree)), bstinsert(t->rchild,k)
3. 3. p->next=s
4. 4. head->rlink, p->llink
5. 5. CABD
6. 6. 1, 16
7. 7. 0
8. 8. (13, 27, 38, 50, 76, 49, 65, 97)
9. 9. n-1
10. 10. 50

#### 四、算法设计题

1. 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。

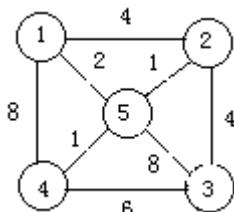
```
void countnode(bitree *bt,int &count)
{
    if(bt!=0)
        {count++; countnode(bt->lchild,count); countnode(bt->rchild,count);}
}
```
2. 2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

```
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;
    }
}
```

### 数据结构试卷 (九)

#### 一、选择题(30分)

1. 下列程序段的时间复杂度为( )。  
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( )个元素。  
(A) n-i (B) n+l-i (C) n-1-i (D) i
3. 设F是由T1、T2和T3三棵树组成的森林，与F对应的二叉树为B，T1、T2和T3的结点数分别为N1、N2和N3，则二叉树B的根结点的左子树的结点数为( )。



### 三、判断题(20分)

1. 1. 有向图的邻接表和逆邻接表中表结点的个数不一定相等。( )  
2. 2. 对链表进行插入和删除操作时不必移动链表中结点。( )  
3. 3. 子串“ABC”在主串“AABCABCD”中的位置为2。( )

4. 4. 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。（ ）
5. 5. 希尔排序算法的时间复杂度为  $O(n^2)$ 。（ ）
6. 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。（ ）
7. 7. 中序遍历一棵二叉排序树可以得到一个有序的序列。（ ）
8. 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。（ ）
9. 9. 顺序表查找指的是在顺序存储结构上进行查找。（ ）
10. 10. 堆是完全二叉树，完全二叉树不一定是堆。（ ）

### 五、算法设计题(20分)

1. 1. 设计计算二叉树中所有结点值之和的算法。
2. 2. 设计将所有奇数移到所有偶数之前的算法。
3. 3. 设计判断单链表中元素是否是递增的算法。

### 数据结构试卷（九）参考答案

#### 一、选择题

1. A 2. A 3. A 4. C 5. D  
 6. D 7. C 8. B 9. C 10. A  
 11. C 12. C 13. D 14. A 15. A

#### 二、填空题

1. 1. p->next, s->data  
 2. 2. 50  
 3. 3. m-1  
 4. 4. 6, 8  
 5. 5. 快速, 堆  
 6. 6. 19/7  
 7. 7. CBDA  
 8. 8. 6  
 9. 9. (24, 65, 33, 80, 70, 56, 48)  
 10. 10. 8

#### 三、判断题

1. 错 2. 对 3. 对 4. 对 5. 错  
 6. 错 7. 对 8. 对 9. 错 10. 对

#### 四、算法设计题

1. 1. 设计计算二叉树中所有结点值之和的算法。

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

2. 2. 设计将所有奇数移到所有偶数之前的算法。

```
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. 3. 设计判断单链表中元素是否是递增的算法。

```
int isrise(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);
}
```

### 数据结构试卷（十）

#### 一、选择题(24分)

1. 下列程序段的时间复杂度为( )。  
 $i=0, s=0; \text{while } (s < n) \{s=s+i; i++; \}$   
(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( )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( )。  
(A)  $s->\text{next}=p->\text{next}; p->\text{next}=s;$       (B)  $q->\text{next}=s; s->\text{next}=p;$   
(C)  $p->\text{next}=s->\text{next}; s->\text{next}=p;$       (D)  $p->\text{next}=s; s->\text{next}=q;$
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( )。  
(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( )。  
(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( )个叶子结点。

$$\sum_{i=1}^m (i-1)N_i \quad \sum_{i=1}^m N_i \quad \sum_{i=2}^m N_i \quad 1 + \sum_{i=2}^m (i-1)N_i$$

(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( )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( )。  
(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( )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，则这棵二叉中共有( )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( )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( )。  
(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(48 分，其中最后两小题各 6 分)

1. 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\_\_\_\_\_次，至多需要比较\_\_\_\_\_次。
2. 快速排序算法的平均时间复杂度为\_\_\_\_\_，直接插入排序算法的平均时间复杂度为\_\_\_\_\_。
3. 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\_\_\_\_\_次。
4. 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\_\_\_\_\_个，比较两次查找成功有结点数有\_\_\_\_\_个。
5. 设一棵 m 叉树的结点数为 n，用多重链表表示其存储结构，则该树中有\_\_\_\_\_个空指针域。
6. 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：  
 $q=p->\text{next}; p->\text{data}=q->\text{data}; p->\text{next}=_____;$   $\text{free}(q);$
7. 数据结构从逻辑上划分为三种基本类型：\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
8. 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的时间复杂度为\_\_\_\_\_；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\_\_\_\_\_。
9. 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\_\_\_\_\_。
10. 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\_\_\_\_\_。
11. 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\_\_\_\_\_。

12. 12. 设有向图G中的一条有向边的集合  
 $E=\{<1, 2>, <2, 3>, <1, 4>, <4, 5>, <5, 3>, <4, 6>, <6, 5>\}$ , 则该图的一个拓扑序列为\_\_\_\_\_
13. 13. 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;_____;}bitree;
void createbitree(bitree *&bt)
{
    scanf("%c",&ch);
    if(ch=='#')_____;else
    { bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch; _____;createbitree(bt->rchild);}
}
```
14. 14. 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;} lklist;
void lklistcreate(_____*&head )
{
    for (i=1;i<=n;i++)
    {
        p=(lklist *)malloc(sizeof(lklist));scanf("%d",&(p->data));p->next=0;
        if(i==1)head=q=p;else {q->next=p;_____;}
    }
}
```

### 三、算法设计题(22分)

1. 1. 设计在链式存储结构上合并排序的算法。  
 2. 2. 设计在二叉排序树上查找结点X的算法。  
 3. 3. 设关键字序列( $k_1, k_2, \dots, k_{n-1}, x$ )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。

### 数据结构试卷 (十) 参考答案

#### 一、选择题

1. A 2. D 3. B 4. B 5. B 6. D  
 7. A 8. D 9. D 10. C 11. B 12. D

#### 二、填空题

1. 1. 4, 10  
 2. 2.  $O(n\log_2 n)$ ,  $O(n^2)$   
 3. 3. n  
 4. 4. 1, 2  
 5. 5.  $n(m-1)+1$   
 6. 6. q->next  
 7. 7. 线性结构, 树型结构, 图型结构  
 8. 8.  $O(n^2)$ ,  $O(n+e)$   
 9. 9. 8/3  
 10. 10. (38, 13, 27, 10, 65, 76, 97)  
 11. 11. (10, 13, 27, 76, 65, 97, 38)  
 12. 12. 124653  
 13. 13. struct node \*rchild, bt=0, createbitree(bt->lchild)  
 14. 14. lklist, q=p

#### 三、算法设计题

1. 1. 设计在链式存储结构上合并排序的算法。  

```
void mergelklist(lklist *ha,lklist *hb,lklist *&hc)
{
    lklist *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. 2. 设计在二叉排序树上查找结点X的算法。  

```
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. } 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。  
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;  
}