

中国科学技术大学

2009-2010 学年第一 学期考试试卷

考试科目: 数据结构及其算法 得分: _____

学生所在系: _____ 姓名: _____ 学号: _____

总分		题号	一	二	三	四
合分人		得分				

注意: 所有答题必须写在试卷上, 每张试卷都必须写上姓名和学号。

装订线, 答题时不要超过此线

一、判断题 (对的打 '√', 错的打 '×', 每题 1 分, 共 10 分)

题号	
得分	
评阅人	

1. 采用顺序存储结构表示的线性表可以随机地访问任一数据元素。√ (√)
2. 栈和队列都是限制存取位置的线性结构。√ (√)
3. 不管原始数据的初始排列状态如何, 快速排序都是速度最快的一种排序方法。× (×)
4. 若输入序列为 1, 2, 3, 4, 5, 6, 则通过一个栈可以输出序列 1, 3, 2, 5, 6, 4。√ (√)
5. 当要求时间复杂度为 $O(n \log n)$ 的稳定排序算法时, 可以采用堆排序或归并排序。× ()
6. 在二叉树的先序序列中, 若结点 A 在结点 B 之前, 则结点 A 一定是结点 B 的祖先。× (×)
7. 如果一个完全二叉树的按层次遍历序列是大顶堆 (最大堆), 则在该二叉树中每个结点的值均 \geq 其所有子孙的值。 (√)
8. 图 G 有 n 个顶点和 e 条边。用邻接矩阵表示法存储图 G 时, 存储空间的大小只取决于 n, 与 e 无关。√ (√)
9. 散列表的基本思想是用记录的关键码的值来确定记录的存储地址。√ ()
10. 哈希表的存储空间越大, 查找效率越高。√ ()

二、选择和填空 (1-6 每题 1 分, 7-10 每空 2 分, 共 20 分)

题号	三
得分	
评阅人	

1. 设有一个带头结点的非空双向循环链表, 指针域 prior 指向结点的直接前趋, 指针域 next 指向直接后继; 设指针 P 指向链表中的一个结点, 指针 Q 指向一个待插入的新结点; 能正确完成将 *Q 插 *P 之前的语句组是 C。

- A. $Q \rightarrow \text{prior} = P \rightarrow \text{prior}; Q \rightarrow \text{next} = P; P \rightarrow \text{prior} = Q; P \rightarrow \text{prior} \rightarrow \text{next} = Q;$
 B. $P \rightarrow \text{prior} = Q; Q \rightarrow \text{next} = P; P \rightarrow \text{prior} \rightarrow \text{next} = Q; Q \rightarrow \text{prior} = P;$
 C. $Q \rightarrow \text{prior} = P \rightarrow \text{prior}; Q \rightarrow \text{prior} \rightarrow \text{next} = Q; P \rightarrow \text{prior} = Q; Q \rightarrow \text{next} = P;$
 D. 前三者都不正确

2. 对由 $n(n > 0)$ 种不同字符组成的电文采用二进制赫夫曼编码, 每个字符编码的长度最长可达满

A 个二进制位。

- A. $n-1$ B. $n/2$ C. $n+1$ D. n

3. 若串 $S1 = "ABCDEFC"$, $S2 = "123"$, 则执行 P 操作的结果是 _____
 $\text{concat}(\text{replace}(S1, \text{substring}(S1, \text{index}(S1, 'E'), 1, 3), S2), "XY")$

- A. ABCD123XY B. XYABC123G C. ABC123GXY D. ABCXY123

4. 下列算法中经一趟排序未必能选出一个元素放在其最终位置的是 _____

- A. 选择 B. 起泡 C. 堆 D. 归并

5. 以数组 $A[0..m-1]$ 实现为循环队列, 其头尾指针分别为 front 和 rear, 从队中最多存放 $m-1$ 个元素, 则队列中的元素个数为 A。

- A. $(\text{rear} - \text{front} + m) \% m$ B. $\text{rear} - \text{front} + 1$ C. $(\text{front} - \text{rear} + m) \% m$ D. $(\text{rear} - \text{front}) \% m$

6. 对稀疏矩阵进行压缩存储目的是 _____

- A. 便于运算 B. 便于输入和输出 C. 节省存储空间 D. 降低运算的时间复杂度

7. 如果二叉排序树上包含了关键字值分别 (16, 28, 40, 37, 19, 30) 等多个结点, 下列序列中 C 不可能是在二叉排序树上查找关键字值等于 37 的结点而得到的比较序列。

- A. 16, 40, 28, 37 B. 19, 28, 40, 37 C. 40, 30, 28, 37 D. 40, 16, 30, 37

8. 若以 {2, 3, 6, 7, 10} 作为叶子结点的权值构造哈夫曼树, 则其带权路径长度是 28。

9. 已函数如下 $\text{int } f(\text{int } x) \{ \text{return } ((x \neq 0) ? x + f(x-1) : 2); \}$ 则 $f(f(1))$ 的返回值是 4。

10. 对有 890 个结点的完全二叉树 T 按自上而下、自左向右的次序编号, 树根的编号为 1, 111 号结点位于 T 的 9 层, 是 T 9 子树上的结点; 以 111 号结点为根的子树上有 4 个结点; 中序遍历以 111 号结点为根的子树时, 最后被访问的结点的编号是 _____。

(注: $2^1=16, 2^2=32, 2^3=64, 2^4=128, 2^5=256, 2^6=512, 2^7=1024$)

11
69
97
128

810
512
378
7

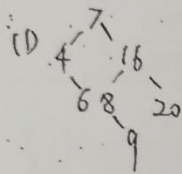
$$f(1) = 2 \quad x * f(0) \quad 2 * f(0) = 2 * f(0)$$

$$f(0) = 2$$

3. 依次读入给定的整数序列 (7, 16, 4, 8, 20, 9, 6); 完成下列操作: (10分)

1) 构造一棵二叉排序树, 计算在等概率情况下该二叉排序树的平均查找长度 ASL;

2) 若变更序列中元素的排列, 可构造出平均查找长度达到最小的二叉排序树, 写出满足上述要求的序列中的第一个元素。

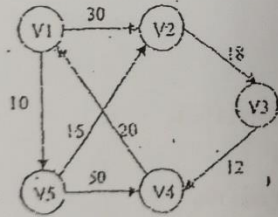


$$\frac{1}{7}(1 \times 2 + 2 \times 2 + 3 \times 3 + 4)$$

$$= \frac{1}{7}(1 + 4 + 9 + 4) = \frac{18}{7}$$

(2).

4. 现有一有向图如下图所示, 求从顶点 V_1 到顶点 V_4 的最短路径及其长度, 要求画出求解过程。(10分)



	S	D	P
V_1	1	0	
V_2	0	30	1, 2
V_3	0	∞	
V_4	0	∞	
V_5	0	10	1, 5

	S	P	P
V_1	1	0	
V_2	0	30	1, 2
V_3	0	20	
V_4	0	60	1, 5, 4
V_5	0	10	1, 5

	S	D	P
V_1	1	0	
V_2	1	35	1, 5, 2
V_3	0	18	1, 5, 2, 3
V_4	0	58	1, 5
V_5	1	10	1, 5

姓名

学号

装订线 答题时不要超过此线

5. 已知以二叉链表作二叉树的存储结构, 阅读算法 f5, 并回答问题: (8分)

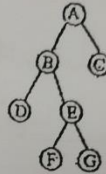
(1) 设二叉树 T 如图所示, 写出执行 f5(T) 的返回值;

(2) 简述算法 f5 的功能。

```

int f5(BiTree T)
{
    int m, n;
    if(!T) return 0;
    else{
        m=f5(T->lchild);
        n=f5(T->rchild);
        if(m>n) return m+1;
        else return n+1;
    }
}

```



- (1) 返回值为 4
 (2) 求二叉树的深度

四、算法设计 (共 20 分)

1. 设一棵二叉树以二叉链表为存储结构, 结点结构为 (lchild, data, rchild), 设计一个算法求二叉树中层次数为 L 的结点数。(5分)

```

int getnum(BiTree &T)

```

```

{ if(!T) return 0;

```

```

  int n;

```

```

  n = (getnum(T->lchild) + getnum(T->rchild)) + 1;

```

```

  return n; }

```

题号	四
得分	
评阅人	

2. 已知不带头结点的单链表L，链表中结点结构为 (data, next)，其中 data 为数据域，next 为指针域。请写一算法，将该链表按结点数据域值的大小从小到大重新链接。要求处理过程中不得新开辟任何结点空间。(11分)

中国科学技术大学

2008-2009 学年第一 学期考试试卷

考试科目: 数据结构及其算法 得分: _____

学生所在系: _____ 姓名: _____ 学号: _____

总分		题号	一	二	三
合分人		得分			

一、选择和填空 (选择每题 1 分, 填空每题 2 分, 共 26 分)

题号	一
得分	
评阅人	

1. 按增长率升序排列函数 2^{100} , $(2/3)^n$, n^4 , n , $n^{3/2}$, $\log_2 n$, $n \log_2 n$: 2^{100} , $\log_2 n$, n , $n \log_2 n$, $n^{3/2}$, n^4 , $(2/3)^n$
2. n 个结点的完全二叉树中, 叶子结点数是 $\lceil \frac{n+1}{2} \rceil$
3. 如果进站的车厢序列为 123, 则可能得到的出站车厢序列是 123, 132, 231, 213, 321
4. 对 n 个结点的单链表查找值为 x 的结点; 在查找成功的情况下, 需平均比较 $\frac{n+1}{2}$ 个结点.
5. 树高为 h 的二叉树上只有度为 0 和 2 的结点, 则它所包含的结点数至少有 $2^h - 1$ 个结点.
6. 已知一个图的邻接矩阵表示, 计算第 i 个结点的入度方法是 计算邻接矩阵主对角线以下元素之和
7. 有一个长度为 12 的有序表, 按二分查找法对该表进行查找, 在表内各元素等概率情况下, 查找成功所需的平均比较次数为 $67/12$.
8. 已知 5 个字符的权值分别为 4, 2, 3, 7, 5, 构造相应哈夫曼树, 其带权路径长度 (WPL) 值为 56.
9. 按照二叉树的定义, 具有 3 个结点的二叉树有 (C) 种.
(A) 3 (B) 4 (C) 5 (D) 6
10. 某二叉树的前序遍历结点访问顺序是 abdgecfh, 中序遍历结点访问顺序是 dgbacfh, 则其后序遍历结点访问顺序是 ().
(A) bdgcefha (B) gdbecfha (C) bdgaecfh (D) gdbefhca
11. 图的深度优先遍历算法类似于二叉树的 (A).
(A) 先序遍历 (B) 中序遍历 (C) 后序遍历 (D) 按层遍历
12. 判断一个有向图是否存在回路, 除了可以利用拓扑排序方法外, 还可以利用 ().
(A) 求关键路径方法 (B) 求最短路径方法 (C) 宽度优先遍历算法 (D) 深度优先遍历方法
13. 线性表若采取链式存储结构时, 要求内存中可用存储单元的地址 (D).
(A) 必须是连续的 (B) 部分地址必须是连续的 (C) 一定是不连续的 (D) 连续与否都行
14. 设有 1000 个无序元素, 希望用最快的速度挑选出其中前 3 个最大的元素, 最好选用 ().
(A) 起泡排序 (B) 快速排序 (C) 堆排序 (D) 归并排序

装订线 答题时不要超过此线

15. 在待排序的元素序列基本有序的前提下, 效率最高的排序方法是 ()。
 (A) 插入排序 (B) 选择排序 (C) 快速排序 (D) 归并排序
16. 关于 n 个结点连通网的最小生成树描述错误的是 ()。
 (A) 含 $n-1$ 条边 (B) 没有回路 (C) 含权值最小的 $n-1$ 个边 (D) 含 n 个结点
17. 设有两个串 p 和 q , 求 q 在 p 中首次出现的位置的运算称作 ()。
 (A) 连接 (B) 模式匹配 (C) 求子串 (D) 求串长
18. 在具有 n 个单元的循环队列中, 队满时共有 () 个元素。
 (A) $n/2$ (B) $n-1$ (C) $\log_2 n$ (D) n^2

三、应用题 (共五题, 50 分)

题号	二
得分	
评阅人	

1. 给出一组关键字 {14, 3, 17, 32, 9, 31, 6, 12, 24, 7, 20} 写出用下列算法从小到大排序时第一趟结束时的序列:
 1) 起泡排序 2) 归并排序 3) 基数排序 (9 分)
2. 按照低下标优先存储的原则存储整数数组 $A[8][7][6][5]$, 每个整数占用 4 个字节, 假设第一元素的地址是 100, 试计算下列几个元素的存储地址: a_{2009} , a_{1111} , a_{1234} . (6 分)
3. 已知后缀表达式为 "abcdef*/-g+h-", 其中所有运算符都是双目算符. (10 分)
 (1) 画出等价的表达式二叉树;
 (2) 写出等价的、包含括号的中缀表达式;
 (3) 用表达式二叉树的树根和一个空栈 S 为实参, 调用下面算法, 写出算法执行后全局变量 m 的值.
- ```
int m=0; //全局变量
void cal(BiTree T, Stack &S){
 if(!T) return;
 push(S,T);
 if(m<StackLength(S)) m=StackLength(S);
 cal(T->lchild, S);
 cal(T->rchild, S);
 Pop(S);
}
```

4. AOE 网  $G$  及其存储结构如图 1 所示, 其存储类型定义如下: (14 分)

```
#define N 20
typedef struct{
 int arcs[N][N]; //邻接矩阵
 char vexs[N]; //存放顶点的数组
 int vexnum, arcnum, kind; // 顶点数、边数、图的类型
}MGraph;
```



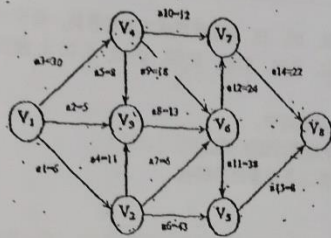


图1

| G.vexs |                | 0              | 1              | 2              | 3              | 4              | 5              | 6              | 7 |
|--------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|---|
|        | V <sub>1</sub> | V <sub>2</sub> | V <sub>3</sub> | V <sub>4</sub> | V <sub>5</sub> | V <sub>6</sub> | V <sub>7</sub> | V <sub>8</sub> |   |
| G.arcs | 0              | 1              | 2              | 3              | 4              | 5              | 6              | 7              |   |
| 0      | ∞              | 6              | 5              | 30             | ∞              | ∞              | ∞              | ∞              |   |
| 1      | ∞              | ∞              | 11             | ∞              | 43             | 6              | ∞              | ∞              |   |
| 2      | ∞              | ∞              | ∞              | ∞              | ∞              | 12             | ∞              | ∞              |   |
| 3      | ∞              | ∞              | ∞              | 8              | ∞              | ∞              | 18             | 12             |   |
| 4      | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | 8              |   |
| 5      | ∞              | ∞              | ∞              | ∞              | ∞              | 38             | ∞              | 24             |   |
| 6      | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | 22             |   |
| 7      | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              | ∞              |   |

求解下列问题:

(1) 用G和int v=0 做实参, 执行算法调用DFS1(G, v); 写出算法的输出序列。

```
int visited[N]={0}; //全局数组
void DFS1(MGraph G, int v){
 int i;
 printf(G.vexs[v]);
 visited[v]=1;
 for(i=0; i<G.vexnum; i++)
 if(!visited[i] && G.arcs[v][i]!=∞) DFS1(G, i);
} //DFS1
```

(2) 用G和int v=0 做实参, 执行算法调用DFS2(G, v); 写出算法的输出序列。

```
int visited[N]={0}; //全局数组
void DFS2(MGraph G, int v){
 int i;
 visited[v]=1;
 for(i=0; i<G.vexnum; i++)
 if(!visited[i] && G.arcs[v][i]!=∞) DFS2(G, i);
 printf(G.vexs[v]);
} //DFS2
```

(3) 写出AOE网中每个顶点的最早发生时间和最迟发生时间; 每个活动的最早开始时间和最迟开始时间; 写出AOE网的关键路径; 哪些活动提前1天完成将导致整个工程提前一天完成。

每个顶点的最早发生时间 ve 和最迟发生时间 vl

| 顶点 | V <sub>1</sub> | V <sub>2</sub> | V <sub>3</sub> | V <sub>4</sub> | V <sub>5</sub> | V <sub>6</sub> | V <sub>7</sub> | V <sub>8</sub> |
|----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| ve |                |                |                |                |                |                |                |                |
| vl |                |                |                |                |                |                |                |                |

每个活动的最早开始时间  $e(i)$  和最迟开始时间  $l(i)$

| 活动     | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ | $a_{13}$ | $a_{14}$ |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|
| $e(i)$ |       |       |       |       |       |       |       |       |       |          |          |          |          |          |
| $l(i)$ |       |       |       |       |       |       |       |       |       |          |          |          |          |          |

5. 有 10 个元素 {26, 14, 23, 01, 66, 20, 55, 05, 18, 09} 欲形成二叉检索树, 每个元素的查找概率对应为 {0.25, 0.1, 0.08, 0.02, 0.1, 0.14, 0.12, 0.06, 0.08, 0.05}, 按照某种次序插入二叉检索树后形成图 2。(11 分)
- (1) 请将这 10 个元素填写到下图所示的二叉检索树中。
  - (2) 求这棵二叉检索树在查找成功情况下的平均查找长度 ASL。
  - (3) 现要删除值为 18 的元素, 画出删除操作后二叉检索树的形状。

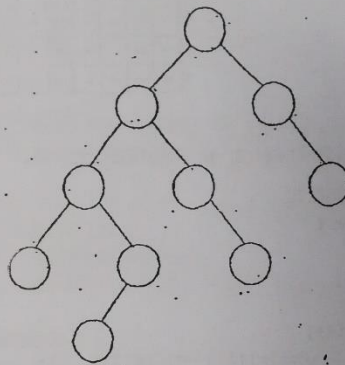


图 2

### 三、算法设计 (共 24 分)

1. 利用两个栈 S1 和 S2 来模拟一个队列。若不存在栈溢出问题, 则请写出用栈的操作来实现队列的入队列 EnQueue 和出队列 DeQueue 的算法。(8 分)

|     |   |
|-----|---|
| 题号  | 三 |
| 得分  |   |
| 评阅人 |   |

2. 现有一个单链表 (每个结点包含 data 域和 next 域, 不带头结点), 其头指针为 phead, 假设其内容已从小到大排好序。(12 分)
  - (1) 完成一个折半查找算法 BiSearch(phead, key, low, high), 若找到则返回相应的序号, 否则返回 -1。要求空间复杂度为  $O(1)$ 。(8 分)
  - (2) 若找不到, 则将 key 插入到其中适当位置, 使得该链表仍然有序。(4 分)
3. 编写一个算法, 判断一个二叉树是否是完全二叉树。设二叉树的根为 T, 每个结点的三个域分别是: data, lchild 和 rchild。(4 分)

《数据结构》试卷 A (开一页)

站点 \_\_\_\_\_ 专业年级 \_\_\_\_\_ 姓名 \_\_\_\_\_ 学号 \_\_\_\_\_ 成绩 \_\_\_\_\_

一、填空题 (每空 1 分, 共 22 分)

1. 数据结构被形式地定义为  $(D, R)$ , 其中  $D$  是 \_\_\_\_\_ 的有限集合,  $R$  是  $D$  上的 \_\_\_\_\_ 有限集合。
2. 一个算法的效率可分为 \_\_\_\_\_ 效率和 \_\_\_\_\_ 效率。
3. 向一个长度为  $n$  的向量的第  $i$  个元素 ( $1 \leq i \leq n+1$ ) 之前插入一个元素时, 需向后移动 \_\_\_\_\_ 个元素。
4. 在一个循环队列中, 队首指针指向队首元素的 \_\_\_\_\_ 位置。
5. 在具有  $n$  个单元的循环队列中, 队满时共有 \_\_\_\_\_ 个元素。
6. 向栈中压入元素的操作是先 \_\_\_\_\_, 后 \_\_\_\_\_。
7. \_\_\_\_\_ 称为空串; \_\_\_\_\_ 称为空白串。
8. 假设有二维数组  $A_{6 \times 8}$ , 每个元素用相邻的 6 个字节存储, 存储器按字节编址。已知  $A$  的起始存储位置 (基地址) 为 1000, 则数组  $A$  的体积 (存储量) 为 \_\_\_\_\_; 末尾元素  $A_{57}$  的第一个字节地址为 \_\_\_\_\_; 若按行存储时, 元素  $A_{14}$  的第一个字节地址为 \_\_\_\_\_; 若按列存储时, 元素  $A_{47}$  的第一个字节地址为 \_\_\_\_\_。
9. 设一棵完全二叉树具有 1000 个结点, 则此完全二叉树有 \_\_\_\_\_ 个叶子结点, 有 \_\_\_\_\_ 个度为 2 的结点, 有 \_\_\_\_\_ 个结点只有非空左子树, 有 \_\_\_\_\_ 个结点只有非空右子树。
10. 线性有序表  $(a_1, a_2, a_3, \dots, a_{216})$  是从小到大排列的, 对一个给定的值  $k$ , 用二分法检索表中与  $k$  相等的元素, 在查找不成功的情况下, 最多需要检索 \_\_\_\_\_ 次。设有 100 个结点, 用二分法查找时, 最大比较次数是 \_\_\_\_\_。
11. 散列法存储的基本思想是由 \_\_\_\_\_ 决定数据的存储地址。

二、判断题 (每题 1 分, 共 10 分)

- ( ) 1. 队是一种插入与删除操作分别在表的两端进行的线性表, 是一种先进后出型结构。
- ( ) 2. 二叉树中所有结点个数是  $2^k - 1$ , 其中  $k$  是树的深度。
- ( ) 3. 栈和队列的存储方式既可是顺序方式, 也可是链接方式。
- ( ) 4. 二叉树中所有结点, 如果不存在非空左子树, 则不存在非空右子树。
- ( ) 5. 对于一棵非空二叉树, 它的根结点作为第一层, 则它的第  $i$  层上最多能有  $2^{i-1}$  个结点。
- ( ) 6. 链表的删除算法很简单, 因为当删除链中某个结点后, 计算机会自动将后续各个单元向前移动。
- ( ) 7. 用二叉链表法 (link-rlink) 存储包含  $n$  个结点的二叉树, 结点的  $2n$  个指针区域中有  $n+1$  个为空指针。
- ( ) 8. 具有 12 个结点的完全二叉树有 5 个度为 2 的结点。
- ( ) 9. 线性表在顺序存储时, 逻辑上相邻的元素未必在存储的物理位置次序上相邻。
- ( ) 10. 顺序表结构适宜于进行顺序存取, 而链表适宜于进行随机存取。

三、单项选择题 (每小题 2 分, 共 18 分)

- ( ) 1. 数据在计算机存储器内表示时, 物理地址与逻辑地址相同并且是连续的, 称之为:  
(A) 存储结构 (B) 逻辑结构 (C) 顺序存储结构 (D) 链式存储结构

( ) 2. 一个向量第一个元素的存储地址是 100, 每个元素的长度为 2, 则第 5 个元素的地址是\_\_\_\_\_

- (A) 110 (B) 108 (C) 100 (D) 120

( ) 3. 在  $n$  个结点的顺序表中, 算法的时间复杂度是  $O(1)$  的操作是:

- (A) 访问第  $i$  个结点 ( $1 \leq i \leq n$ ) 和求第  $i$  个结点的直接前驱 ( $2 \leq i \leq n$ )  
 (B) 在第  $i$  个结点后插入一个新结点 ( $1 \leq i \leq n$ )  
 (C) 删除第  $i$  个结点 ( $1 \leq i \leq n$ ) (D) 将  $n$  个结点从小到大排序

( ) 4. 向一个有 127 个元素的顺序表中插入一个新元素并保持原来顺序不变, 平均要移动\_\_\_\_\_个元素

- (A) 8 (B) 63.5 (C) 63 (D) 7

( ) 5. 判定一个队列 QU (最多元素为  $m0$ ) 为满队列的条件是

- A.  $QU \rightarrow rear - QU \rightarrow front == m0$  B.  $QU \rightarrow rear - QU \rightarrow front - 1 == m0$  C.  $QU \rightarrow front == QU \rightarrow rear$  D.  $QU \rightarrow front == QU \rightarrow rear + 1$

( ) 6. 链表是一种采用\_\_\_\_\_存储结构存储的线性表:

- (A) 顺序 (B) 链式 (C) 星式 (D) 网状

( ) 7. 线性表若采用链式存储结构时, 要求内存中可用存储单元的地址:

- (A) 必须是连续的 (B) 部分地址必须是连续的  
 (C) 一定是不连续的 (D) 连续或不连续都可以

( ) 8. 线性表 L 在\_\_\_\_\_情况下适用于使用链式结构实现。

- (A) 需经常修改 L 中的结点值 (B) 需不断对 L 进行删除插入  
 (C) L 中含有大量的结点 (D) L 中结点结构复杂

( ) 9. 若已知一个栈的入栈序列是 1, 2, 3, ...,  $n$ , 其输出序列为  $p_1, p_2, p_3, \dots, p_n$ , 若  $p_1 = n$ , 则  $p_i$  为

- A.  $i$  B.  $n-i$  C.  $n-i+1$  D. 不确定

#### 四、简答题 (10 分)

1. 已知如图所示的有向图, 请给出该图的:



- (1) 每个顶点的入/出度;  
 (2) 邻接矩阵;  
 (3) 邻接表;  
 (4) 逆邻接表。

| 顶点 | 1 | 2 | 3 | 4 | 5 | 6 |
|----|---|---|---|---|---|---|
| 入度 |   |   |   |   |   |   |
| 出度 |   |   |   |   |   |   |

2. 线性表具有两种存储方式, 即顺序方式和链接方式。现有一个具有五个元素的线性表  $L = \{23, 17, 47, 05, 31\}$ , 若它以链接方式存储在下列 100~119 号地址空间中, 每个结点由数据 (占 2 个字节) 和指针 (占 2 个字节) 组成, 如下所示:

|    |   |    |   |    |   |    |   |    |   |
|----|---|----|---|----|---|----|---|----|---|
| 05 | U | 17 | X | 23 | V | 31 | Y | 47 | Z |
|----|---|----|---|----|---|----|---|----|---|

100

120

其中指针 X, Y, Z 的值分别为多少? 该线性表的首结点起始地址为多少? 末结点的起始地址为多少?

五、写出下列程序段的输出结果（栈的元素类型 SElem Type 为 char）。

(1)

```
void main(){
 Stack S;
 char x,y;
 InitStack(S);
 x='c';y='k';
 push(S,x); push(S,'a'); push(S,y);
 pop(S,x); push(S,'t'); push(S,x);
 pop(S,x); push(S,'s');
 while(!StackEmpty(S)){ pop(S,y);printf(y); };
 printf(x);
}
```

结果: \_\_\_\_\_

(2) 写出下列程序段的输出结果（队列中的元素类型 QElem Type 为 char）。

```
void main(){
 Queue Q; Init Queue (Q);
 Char x='e'; y='e';
 EnQueue (Q,'h'); EnQueue (Q,'r'); EnQueue (Q,'y');
 DeQueue (Q,x); EnQueue (Q,x);
 DeQueue (Q,x); EnQueue (Q,'a');
 while(!QueueEmpty(Q)){ DeQueue (Q,y);printf(y); };
 Printf(x);
}
```

结果: \_\_\_\_\_

6. 假设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。试为这 8 个字母设计哈夫曼编码。使用 0~7 的二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。

五、算法设计（在下列算法的横线内填上适当的语句或表达式。）

1. 直接选择排序

```
void selectsort (int R[])
// 按递增序对 R[0] ~ R[n-1] 进行直接选择排序
{ int i, j, k, temp;
 for (i=0; i<= _____ ; i++)
 { k=i;
 for (j= _____ ; j<=n-1; j++)
 if (R[j] _____ R[k])
 k=j;
```

```

 if (_____)
 { temp=R[i]; R[i] = _____; R[k]=temp; }
 }
}

```

2. 中序遍历二叉树

设二叉树用二叉链表表示，以 t 为根指针，二叉链表结点的类型为 node；栈 s 的元素类型为指向 node 的指针类型，栈容量 m 足够大，中序遍历的非递归算法如下：

```

struct node
{
 char data;
 node *lc, *rc;
};

void preorder (node *t)
{
 node *s[m], *p=t;

 int top = -1; //置栈空
 do
 {
 while (p!=NULL)
 {
 s[++top] = _____;
 _____;
 }
 if (top != -1)
 {
 p=s[top--];
 _____;
 _____;
 }
 } while ((_____) || (p != NULL));
}

```

一、填空题（每空 1 分，共 22 分）

1. 数据结构按形式地定义为 (D, R)，其中 D 是 数据元素 的有限集合，R 是 D 上的 关系 有限集合。
2. 一个算法的效率可分为 时间 效率和 空间 效率。

```

 if (_____)
 { temp=R[i]; R[i]=_____; R[k]=temp; }
 }
}

```

## 2. 中序遍历二叉树

设二叉树用二叉链表表示，以 t 为根指针，二叉链表结点的类型为 node；栈 s 的元素类型为指向 node 的指针类型，栈容量 m 足够大。中序遍历的非递归算法如下：

```

struct node
{
 char data;
 node *lc, *rc;
};

void preorder (node *t)
{
 node *s[m], *p=t;
 int top = 1; //置栈空
 do
 {
 while (p!=NULL)
 {
 s[++top] = _____;
 _____;
 }
 if (top != -1)
 {
 p=s[top--];
 _____;
 _____;
 }
 } while ((_____) || (p != NULL));
}

```

## 一、填空题 (每空 1 分, 共 22 分)

1. 数据结构被形式地定义为  $(D, R)$ , 其中 D 是 数据元素 的有限集合, R 是 D 上的 关系 有限集合。
2. 一个算法的效率可分为 时间 效率和 空间 效率。

3. 向一个长度为  $n$  的向量的第  $i$  个元素 ( $1 \leq i \leq n+1$ ) 之前插入一个元素时, 需向后移动  $n+1$  个元素。
4. 在一个循环队列中, 队首指针指向队首元素的 前一个 位置。
5. 在具有  $n$  个单元的循环队列中, 队满时共有  $n-1$  个元素。
6. 向栈中压入元素的操作是先 移动栈顶指针, 后 存入元素。
7. 不包含任何字符 (长度为 0) 的串 称为空串; 由一个或多个空格 (仅由空格符) 组成的串 称为空白串。
8. 假设有二维数组  $A_{6 \times 8}$ , 每个元素用相邻的 6 个字节存储, 存储器按字节编址。已知  $A$  的起始存储位置 (基地址) 为 1000, 则数组  $A$  的体积 (存储量) 为 288 B; 末尾元素  $A_{67}$  的第一个字节地址为 1282; 若按行存储时, 元素  $A_{11}$  的第一个字节地址为  $(8+4) \times 6 + 1000 = 1072$ ; 若按列存储时, 元素  $A_{47}$  的第一个字节地址为  $(6 \times 7 + 4) \times 6 + 1000 = 1276$ 。
9. 设一棵完全二叉树具有 1000 个结点, 则此完全二叉树有 500 个叶子结点, 有 499 个度为 2 的结点, 有 1 个结点只有非空左子树, 有 0 个结点只有非空右子树。
10. 线性有序表 ( $a_1, a_2, a_3, \dots, a_{256}$ ) 是从大到小排列的, 对一个给定的值  $k$ , 用二分法检索表中与  $k$  相等的元素, 在查找不成功的情况下, 最多需要检索 8 次。设有 100 个结点, 用二分法查找时, 最大比较次数是 7。
11. 散列法存储的基本思想是由 关键字的值 决定数据的存储地址。

一、判断题 (每题 1 分, 共 10 分)

- () 9. 队是一种插入与删除操作分别在表的两端进行的线性表, 是一种先进后出型结构。
- () 1. 二叉树中所有结点个数是  $2^k - 1$ , 其中  $k$  是树的深度。(应  $2^{k+1}$ )
- () 7. 栈和队列的存储方式既可是顺序方式, 也可是链接方式。
- () 2. 二叉树中所有结点, 如果不存在非空左子树, 则不存在非空右子树。
- () 3. 对于一棵非空二叉树, 它的根结点作为第一层, 则它的第  $i$  层上最多能有  $2^{i-1}$  个结点。(应  $2^{i+1}$ )
- () 3. 链表的删除算法很简单, 因为当删除链中某个结点后, 计算机会自动地将后续的各个单元向前移动。
- () 4. 用二叉链表法 (link-rlink) 存储包含  $n$  个结点的二叉树, 结点的  $2n$  个指针区域中有  $n+1$  个为空指针。
- () 5. 具有 12 个结点的完全二叉树有 5 个度为 2 的结点。
- () 8. 线性表在顺序存储时, 逻辑上相邻的元素未必在存储的物理位置次序上相邻。
- () 5. 顺序表结构适宜于进行顺序存取, 而链表适宜于进行随机存取。

三、单项选择题 (每小题 2 分, 共 18 分)

- () 1. 数据在计算机存储器内表示时, 物理地址与逻辑地址相同并且是连续的, 称之为: (A) 存储结构 (B) 逻辑结构 (C) 顺序存储结构 (D) 链式存储结构
- () 2. 一个向量第一个元素的存储地址是 100, 每个元素的长度为 2, 则第 5 个元素的地址是: (A) 110 (B) 108 (C) 100 (D) 120
- () 3. 在  $n$  个结点的顺序表中, 算法的时间复杂度是  $O(1)$  的操作是: (D) 访问第  $i$  个结点 ( $1 \leq i \leq n$ ) 和求第  $i$  个结点的直接前驱 ( $2 \leq i \leq n$ )



- (E) 在第  $i$  个结点后插入一个新结点 ( $1 \leq i \leq n$ )
- (F) 删除第  $i$  个结点 ( $1 \leq i \leq n$ )
- (G) 将  $n$  个结点从小到大排序

( B ) 4. 向一个有 127 个元素的顺序表中插入一个新元素并保持原来顺序不变, 平均要移动    个元素  
 (A) 8      (B) 63.5      (C) 63      (D) 7

( A ) 4. 判定一个队列 QU (最多元素为  $m0$ ) 为满队列的条件是     
 A.  $QU \rightarrow rear - QU \rightarrow front == m0$     B.  $QU \rightarrow rear - QU \rightarrow front - 1 == m0$   
 C.  $QU \rightarrow front == QU \rightarrow rear$             D.  $QU \rightarrow front == QU \rightarrow rear + 1$

( B ) 6. 链表是一种采用    存储结构存储的线性表;  
 (A) 顺序    (B) 链式    (C) 星式    (D) 网状

( D ) 7. 线性表若采用链式存储结构时, 要求内存中可用存储单元的地址:  
 (A) 必须是连续的    (B) 部分地址必须是连续的  
 (C) 一定是不连续的    (D) 连续或不连续都可以

( B ) 8. 线性表 L 在    情况下适用于使用链式结构实现。  
 (A) 需经常修改 L 中的结点值    (B) 需不断对 L 进行删除插入  
 (C) L 中含有大量的结点    (D) L 中结点结构复杂

( C ) 9. 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列为  $p_1, p_2, p_3, \dots, p_n$ , 若  $p_1 = n$ , 则  $p_i$  为  
 A.  $i$     B.  $n-i$     C.  $n-i+1$     D. 不确定

四、

1. 略

2. 答:  $X = 116$      $Y = 0$      $Z = 100$     首址 = 108    末址 = 112

五、

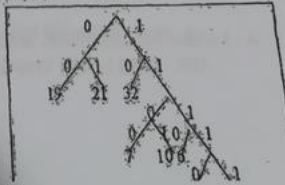
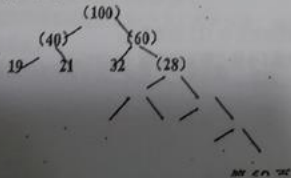
1. 答: 输出为 "stack".

2. 答: 输出为 "char".

六、解: 方案 1: 哈夫曼编码

先将概率放大 100 倍, 以方便构造哈夫曼树。

$w = \{7, 19, 2, 6, 32, 3, 21, 10\}$ , 按哈夫曼规则:  $[ (2, 3), 6 ], (7, 10), \dots, 19, 21, 32$



(17) (11)  
7 10 6 (5)  
2 3

方案比较:

| 字母编号 | 对应编码  | 出现频率 |
|------|-------|------|
| 1    | 1100  | 0.07 |
| 2    | 00    | 0.19 |
| 3    | 11110 | 0.02 |
| 4    | 1110  | 0.06 |
| 5    | 10    | 0.32 |
| 6    | 11111 | 0.03 |

| 字母编号 | 对应编码 | 出现频率 |
|------|------|------|
| 1    | 000  | 0.07 |
| 2    | 001  | 0.19 |
| 3    | 010  | 0.02 |
| 4    | 011  | 0.06 |
| 5    | 100  | 0.32 |
| 6    | 101  | 0.03 |

方案 1 的 WPL =  $2(0.19+0.32+0.21)+4(0.07+0.06+0.10)+5(0.02+0.03)=1.44+0.92+0.25=2.61$

方案 2 的 WPL =  $3(0.19+0.32+0.21+0.07+0.06+0.10+0.02+0.03)=3$

结论: 哈夫曼编码优于等长二进制编码

#### 六、阅读分析题 (10 分)

指出以下算法中的错误和低效(即费时)之处, 并将它改写为一个既正确又高效的算法。

```

Status DeleteK(SqList&a, int i, int k){
//本过程从顺序存储结构的线性表 a 中删除第 i 个元素起的 k 个元素
if (i<1 || k<0 || i+k> a.length) return INFEASIBLE; //参数不合法
else{
for(count = 1; count <k; count ++){
//删除一个元素
for (j = a.length; j>=i+1; j--) a.elem[j-1] = a.elem[j];
a.length --;
}
return OK;
} //DeleteK

```

注: 上题涉及的类型定义如下:

```

#define LIST_INIT_SIZE 100
#define LISTINCREMENT 10
typedef struct {
ElemType *elem; //存储空间基址
int length; //当前长度
int listsize; //当前分配的存储容量
}SqList;

```

应将 `if (i<1 || k<0 || i+k> a.length) return INFEASIBLE`

改为: if(!((0<i<=a.length)^(0<k<=a.length-i))) return INFEASIBLE

第二个 FOR 语句中, 元素前移的次序错误。应将 for (j = a.length; j>=i+1; j--) a.elem[j-1] = a.elem[j];

改为 for (j>=i+1; j = a.length; j++) a.elem[j-1] = a.elem[j];

### 《数据结构》试卷 B (开一页)

站点 \_\_\_\_\_ 专业年级 \_\_\_\_\_ 姓名 \_\_\_\_\_ 学号 \_\_\_\_\_ 成绩 \_\_\_\_\_

#### 一、填空题 (每空 1 分, 共 15 分)

1. 向量、栈和队列都是 线性 结构, 可以在向量的 任意 位置插入和删除元素; 对于栈只能在 尾部 插入和删除元素; 对于队列只能在 尾部 插入和 头部 删除元素。
2. 栈是一种特殊的线性表, 允许插入和删除运算的一端称为 栈顶, 不允许插入和删除运算的一端称为 栈底。
3. 数据结构是一门研究非数值计算的程序设计问题中计算机的 存储结构 以及它们之间的 运算 和 运算等 的学科。
4. 在顺序表中插入或删除一个元素, 需要平均移动 一半 元素, 具体移动的元素个数与 插入或删除元素的位置 有关。
5. 在具有 n 个单元的循环队列中, 队满时共有 n-1 个元素。
6. 假设在有序线性表 a[20] 上进行折半查找, 则比较一次查找成功的结点数为 1; 比较两次查找成功的结点数为 2; 比较四次查找成功的结点数为 4; 平均查找长度为 2.5。

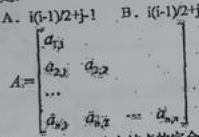
#### 二、判断正误 (判断下列概念的正确性, 并作出简要的说明。) (每小题 1 分, 共 10 分)

- ( ) 1. 线性表的每个结点只能是一个简单类型, 而链表的每个结点可以是一个复杂类型。
- ( ) 2. 在表结构中最常用的是线性表, 栈和队列不太常用。
- ( ) 3. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表, 是一种后进先出型结构。
- ( ) 4. 对于不同的使用者, 一个表结构既可以是栈, 也可以是队列, 也可以是线性表。
- ( ) 5. 线性表的逻辑顺序与存储顺序总是一致的。
- ( ) 6. 栈和队列是一种非线性数据结构。

- ( ) 7. 栈和队列的存储方式既可是顺序方式, 也可是链接方式。
- ( ) 8. 两个栈共享一片连续内存空间时, 为提高内存利用率, 减少溢出机会, 应把两个栈的栈底分别设在这片内存空间的两端。
- ( ) 9. 队是一种插入与删除操作分别在表的两端进行的线性表, 是一种先进后出型结构。
- ( ) 10. 一个栈的输入序列是 12345, 则栈的输出序列不可能是 12345。

三、单项选择题 (每小题 1 分, 共 20 分)

- ( ) 1. 数据在计算机存储器内表示时, 物理地址与逻辑地址相同并且是连续的, 称之为: (A) 存储结构 (B) 逻辑结构 (C) 顺序存储结构 (D) 链式存储结构
- ( ) 2. 若已知一个栈的入栈序列是 1, 2, 3, ..., n, 其输出序列为 p1, p2, p3, ..., pn, 若 p1=n, 则 pi 为  
A. i B. n-i C. n-i+1 D. 不确定
- ( ) 3. 判定一个栈 ST (最多元素为 m0) 为空的条件是  
A. ST->top<0 B. ST->top=0 C. ST->top<m0 D. ST->top=m0
- ( ) 4. 设矩阵 A 是一个对称矩阵, 为了节省存储, 将其下三角部分 (如下图所示) 按行序存放在一维数组 B[1, n(n-1)/2] 中, 对下三角部分中任一元素 a<sub>ij</sub> (i ≤ j), 在一维数组 B 中下标 k 的值是:  
A. i(i-1)/2+j-1 B. i(i-1)/2+j C. i(i+1)/2+j-1 D. i(i+1)/2+j



- ( ) 5. 具有 n (n > 0) 个结点的完全二叉树的深度为 \_\_\_\_\_。  
(A)  $\lceil \log_2(n) \rceil$  (B)  $\lfloor \log_2(n) \rfloor$  (C)  $\lfloor \log_2(n) \rfloor + 1$  (D)  $\lceil \log_2(n) \rceil + 1$
- ( ) 6. 有 8 个结点的无向连通图最少有 \_\_\_\_\_ 条边。  
A. 5 B. 6 C. 7 D. 8

7. 数据结构反映了数据元素之间的结构关系。链表是一种 A, 它对于数据元素的插入和删除 B。通常查找线性表数据元素的方法有 C 和 D 两种方法, 其中 C 是一种只适合于顺序存储结构但 E 的方法; 而 D 是一种对顺序和链式存储结构均适用的方法。

- 供选择的答案:
- A: ①顺序存储线性表 ②非顺序存储非线性表 ③顺序存储非线性表 ④非顺序存储线性表
- B: ①不需要移动结点, 不需改变结点指针 ②不需要移动结点, 只需改变结点指针  
③只需移动结点, 不需改变结点指针 ④既需移动结点, 又需改变结点指针
- C: ①顺序查找 ②循环查找 ③条件查找 ④二分法查找
- D: ①顺序查找 ②随机查找 ③二分法查找 ④分块查找
- E: ①效率较低的线性查找 ②效率较低的非线性查找 ③效率较高的非线性查找 ④效率较高的线性查找

答案: A=\_\_\_\_\_ B=\_\_\_\_\_ C=\_\_\_\_\_ D=\_\_\_\_\_ E=\_\_\_\_\_

8. 散列表存储的基本思想是根据 A 来决定 B, 碰撞(冲突)指的是 C, 处理碰撞的两类主要方法是 D.

供选择的答案

A, B: ① 存储地址 ② 元素的符号 ③ 元素个数 ④ 关键字值  
⑤ 非码属性 ⑥ 平均检索长度 ⑦ 负载因子 ⑧ 散列表空间

C: ① 两个元素具有相同序号 ② 两个元素的关键码值不同, 而非码属性相同  
③ 不同关键码值对应到相同的存储地址 ④ 负载因子过大 ⑤ 数据元素过多

D: ① 线性探查法和双散列函数法 ② 建溢出区法和不建溢出区法  
③ 除余法和折叠法 ④ 拉链法和开地址法

答案: A=\_\_\_\_\_ B=\_\_\_\_\_ C=\_\_\_\_\_ D=\_\_\_\_\_

9. 考虑具有如下性质的二叉树: 除叶子结点外, 每个结点的值都大于其左子树上的一切结点的值, 并小于等于其右子树上的一切结点的值。

现把 9 个数 1, 2, 3, ..., 8, 9 填入下图所示的二叉树的 9 个结点中, 并使它具有上述性质。此时, n1 的值是 A, n2 的值是 B, n9 的值是 C。现欲把  $\sqrt{10}$  放入此树

并使该树保持前述性质, 增加的一个结点可以放在 D 或 E。

供选择的答案

A~C: ① 1 ② 2 ③ 3 ④ 4 ⑤ 5 ⑥ 6  
⑦ 7 ⑧ 8 ⑨ 9

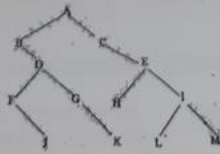
D~E: ① n7 下面 ② n8 下面 ③ n9 下面  
④ n6 下面 ⑤ n1 与 n2 之间 ⑥ n2 与 n4 之间  
⑦ n6 与 n9 之间 ⑧ n3 与 n6 之间

答案: A=\_\_\_\_\_ B=\_\_\_\_\_ C=\_\_\_\_\_  
D=\_\_\_\_\_ E=\_\_\_\_\_



四、简答题 (每小题 5 分, 共 25 分)

1. 说明线性表、栈与队的异同点。
2. 试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列



3. 假设正读和反读都相同的字符序列为“回文”, 例如, ‘abba’ 和 ‘abcba’ 是回文, ‘abcde’ 和 ‘ababab’ 则不是回文。假设一字符序列已存入计算机, 请分析用线性表、堆栈和队列等

六、算法设计 (每小题 5 分, 共 15 分。至少要写出思路)

1. 写出在顺序存储结构下将线性表逆转的算法, 要求使用最少的附加空间。
2. 编写程序, 将若干整数从键盘输入, 以单链表形式存储起来, 然后计算单链表中结点的个数 (其中指针 P 指向该链表的第一个结点)。
3. 试写一个算法, 判别读入的一个以 '@' 为结束符的字符序列是否是“回文”。

一、填空题 (每空 1 分, 共 15 分)

1. 向量、栈和队列都是 线性 结构, 可以在向量的 任何 位置插入和删除元素; 对于栈只能在 栈顶 插入和删除元素; 对于队列只能在 队尾 插入和 队首 删除元素。
2. 栈是一种特殊的线性表, 允许插入和删除运算的一端称为 栈顶, 不允许插入和删除运算的一端称为 栈底。
3. 数据结构是一门研究非数值计算的程序设计问题中计算机的 操作对象 以及它们之间的 关系 和运算等的学科。
4. 在顺序表中插入或删除一个元素, 需要平均移动 表中一半 元素, 具体移动的元素个数与 表长 和 该元素在表中的位置 有关。
5. 在具有 n 个单元的循环队列中, 队满时共有 n-1 个元素。
8. 假设在有序线性表 a[20] 上进行折半查找, 则比较一次查找成功的结点数为 1; 比较两次查找成功的结点数为 2; 比较四次查找成功的结点数为 8; 平均查找长度为 3.7。

解: 显然, 平均查找长度  $= 0(\log_2 n) < 5$  次 ( $2^5$ ), 但具体是多少次, 则不应当按照公式  $ASL = \frac{n+1}{n} \log_2(n+1)$  来计算 (即  $(21 \times \log_2 21) / 20 = 4.6$  次并不正确!), 因为这是在假设  $n = 2^k - 1$  的情况下推导出来的公式。应当用穷举法罗列:  
全部元素的查找次数为  $= (1 + 2 \times 2 + 4 \times 3 + 8 \times 4 + 5 \times 5) = 74$ ;  $ASL = 74 / 20 = 3.7$  !!!

二、判断正误 (判断下列概念的正确性, 并作出简要的说明。) (每小题 1 分, 共 10 分)

- ( X ) 1. 线性表的每个结点只能是一个简单类型, 而链表的每个结点可以是一个复杂类型。  
错, 线性表是逻辑结构概念, 可以顺序存储或链式存储, 与元素数据类型无关。
- ( X ) 2. 在表结构中最常用的是线性表, 栈和队列不太常用。  
错, 不一定吧? 调用子程序或函数常用, CPU 中也用队列。
- (  ) 3. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表, 是一种后进先出型结构。
- (  ) 4. 对于不同的使用者, 一个表结构既可以是栈, 也可以是队列, 也可以是线性表。  
正确, 都是线性逻辑结构, 栈和队列其实是特殊的线性表, 对运算的定义略有不同而已。
- ( X ) 5. 线性表的逻辑顺序与存储顺序总是一致的

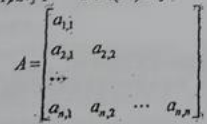
- ( X ) 6. 栈和队列是一种非线性数据结构。  
错，他们都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。
- ( J ) 7. 栈和队列的存储方式既可是顺序方式，也可是链接方式。
- ( J ) 8. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。
- ( X ) 9. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。  
错，后半句不对。
- ( X ) 10. 一个栈的输入序列是 12345，则栈的输出序列不可能是 12345。  
错，有可能。

三、单项选择题（每小题 1 分，共 20 分）

- ( C ) 1. 数据在计算机存储器内表示时，物理地址与逻辑地址相同并且是连续的，称之为：  
(A) 存储结构 (B) 逻辑结构 (C) 顺序存储结构 (D) 链式存储结构
- ( C ) 2. 若已知一个栈的入栈序列是 1, 2, 3, ..., n，其输出序列为 p1, p2, p3, ..., pn，若 p1=n，则 pi 为  
A. i B. n-i C. n-i+1 D. 不确定  
解释：当 p1=n，即 n 是最先出栈的，根据栈的原理，n 必定是最后入栈的（事实上题目已经表明了），那么输入顺序必定是 1, 2, 3, ..., n，则出栈的序列是 n, ..., 3, 2, 1。  
(若不要求顺序出栈，则输出序列不确定)

- ( B ) 3. 判定一个栈 ST（最多元素为 m0）为空的条件是  
A. ST>top<0 B. ST>top=0 C. ST>top<m0  
D. ST>top=m0

- ( B ) 4. 设矩阵 A 是一个对称矩阵，为了节省存储，将其下三角部分（如下图所示）按行序存放在一维数组 B[1, n(n-1)/2] 中，对下三角部分中任一元素 a<sub>ij</sub> (i ≤ j)，在一维数组 B 中下标 k 的值是：  
A. i(i-1)/2+j-1 B. i(i-1)/2+j C. i(i+1)/2+j-1 D. i(i+1)/2+j



- ( C ) 5. 具有 n(n>0) 个结点的完全二叉树的深度为 \_\_\_\_\_。  
(A)  $\lceil \log_2(n) \rceil$  (B)  $\lfloor \log_2(n) \rfloor$  (C)  $\lfloor \log_2(n) \rfloor + 1$  (D)  $\lceil \log_2(n) + 1 \rceil$
- ( C ) 6. 有 8 个结点的无向连通图最少有 \_\_\_\_\_ 条边。  
A. 5 B. 6 C. 7 D. 8

7. 答案: A=④ B=② C=④ D=① E=③  
8. 答案: A=④ B=① C=③ D=④ E=⑥  
9. 答案: A=⑦ B=④ C=⑥ D=② E=⑥

#### 四、简答题 (每小题 4 分, 共 20 分)

##### 1. 说明线性表、栈与队的异同点。

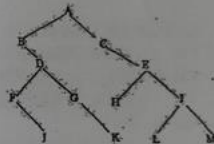
答: 相同点: 都是线性结构, 都是逻辑结构的概念, 都可以用顺序存储或链表存储; 栈和队列是两种特殊的线性表, 即受限的线性表, 只是对插入、删除运算加以限制。

不同点: ① 运算规则不同, 线性表为随机存取, 而栈是只允许在一端进行插入、删除运算, 因而是后进先出表 LIFO; 队列是只允许在一端进行插入、另一端进行删除运算, 因而是先进先出表 FIFO。

② 用途不同, 堆栈用于子程序调用和保护现场, 队列用于多道作业处理、指令寄存及其他运算等等。

##### 2. 试写出如图所示的二叉树分别按先序、中序、后序遍历得到的结点序列。

答: DLR: A B D F J G K C E H I L M  
LDR: B F J D G K A C E H I L M  
LRD: J F K G D B H L M I E C A



##### 3. 假设正读和反读都相同的字符序列为“回文”, 例如, ‘abba’ 和 ‘abcba’ 是回文, ‘abcde’ 和 ‘ababab’ 则不是回文。假设一字符序列已存入计算机, 请分析用线性表、堆栈和队列等方式正确输出其回文的可能性?

答: 线性表是随机存取, 可以实现, 靠循环变量 (j--) 从表尾开始打印输出;

堆栈是后进先出, 也可以实现, 靠正序入栈、逆序出栈即可;

队列是先进先出, 不易实现。

哪种方式最好, 要具体情况具体分析, 若正文在机内已是顺序存储, 则直接用线性表从后往前读取即可, 或将堆栈栈顶开到数组末尾, 然后直接用 POP 动作实现。(但堆栈是先减后压还是……)

若正文是单链表形式存储, 则等同于队列, 需开辅助空间, 可以从链首开始入栈, 全部压入后再依次输出。

##### 4. 试比较顺序存储结构和链式存储结构的优缺点。在什么情况下用顺序表比链表好?

答: ① 顺序存储时, 相邻数据元素的存放地址也相邻 (逻辑与物理统一); 要求内存中可用存储单元的地址必须是连续的。

优点: 存储密度大 ( $=1$ ), 存储空间利用率高, 缺点: 插入或删除元素时不方便。

② 链式存储时, 相邻数据元素可随意存放, 但所占存储空间分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针

优点: 插入或删除元素时很方便, 使用灵活, 缺点: 存储密度小 ( $<1$ ), 存储空间利用率低。

顺序表适宜于做查找这样的静态操作; 链表适宜于做插入、删除这样的动态操作。

若线性表的长度变化不大, 且其主要操作是查找, 则采用顺序表;

若线性表的长度变化较大, 且其主要操作是插入、删除操作, 则采用链表。

##### 5. 给定二叉树的两种遍历序列, 分别是:

前序遍历序列: D, A, C, E, B, H, F, G, I; 中序遍历序列: D, C, B, E, H, A, G,



1. F.

试画出二叉树 B, 并简述由任意二叉树 B 的前序遍历序列和中序遍历序列求二叉树 B 的思路方法。

解: 方法是: 由前序先确定 root, 由中序可确定 root 的左、右子树, 然后由其左子树的元素集合和右子树的集合对应前序遍历序列中的元素集合, 可继续确定 root 的左右孩子, 将他们分别作为新的 root, 不断递归, 则所有元素都被唯一确定, 问题得解。



五、阅读理解 (每小题 5 分, 共 20 分, 至少要写出思路)

答: 此题答案不唯一, 但若从已给定序列中挑选, 则限制颇多。

(7) Q=P;

(11) P=L;

(8) while(P->next!=Q)P=P->next;

(10) P=Q;

(4) S->next=P->next;

P->next=S;

已知 P 结点, 则不必“顺藤摸瓜”, 直接链接即可。

(4) S->next=P->next;

(1) P->next=S;

2. 答: 这是找结点后继的程序。

共有 3 处错误。

注: 当 rtag=1 时说明内装后继指针, 可直接返回, 第一句无错。

当 rtag=0 时说明内装右孩子指针, 但孩子未必是后继, 需要计算。中序遍历应当先左再根再右, 所以应当找左子树直到叶子处, r=r->lchild; 直到 Ltag=1;

应改为: while(!r->Ltag)r=r->lchild;

```
BiTree InSucc(BiTree q){
//已知 q 是指向中序线索二叉树上某个结点的指针,
//本函数返回指向*q 的后继的指针。
r=q->rchild; //应改为 r=q;
if(!r->rtag)
while(!r->rtag)r=r->rchild; //应改为
while(!r->Ltag)r=r->lchild;
return r; //应改为 return r->rchild;
```

3. 写出下列程序段的输出结果 (队列中的元素类型 QElem Type 为 char)。

```
void main()
{
 Queue Q; Init Queue (Q);
 Char x='e'; y='e';
 EnQueue (Q, 'h'); EnQueue (Q, 'r'); EnQueue (Q, 'y');
 DeQueue (Q, x); EnQueue (Q, x);
 DeQueue (Q, x); EnQueue (Q, 'a');
 while(!QueueEmpty(Q)){ DeQueue (Q, y); printf(y); };
 Print(x);
}
```

答: 输出为 "char".

4. 简述以下算法的功能 (栈和队列的元素类型均为 int)。

```
void algo3(Queue &Q){
 Stack S; int d;
 InitStack(S);
 while(!QueueEmpty(Q)){
 DeQueue (Q, d); Push(S, d);
 };
 while(!StackEmpty(S)){
 Pop(S, d); EnQueue (Q, d);
 }
}
```

答: 该算法的功能是: 利用堆栈做辅助, 将队列中的数据元素进行逆置。

六、算法设计 (每小题 5 分, 共 15 分。至少要写出思路)

1. 输入: 长度为 n 的线性表数组 A(1:n)

输出: 逆转后的长度为 n 的线性表数组 A(1:n)。

C 语言描述如下 (其中 ET 为数据元素的类型):

```
invs1(n,a)
{
 int n;
 ET a[];
 (int k;
 ET t;
 for (k=1; k<=n/2; k++)
 {t=a[k-1]; a[k-1]=a[n-k]; a[n-k]=t;}
 return;
}
```

2. 假设一个数组  $sq[m]$  存放循环队列的元素。若要使这  $m$  个分量都得到利用, 则需另一个标志  $tag$ , 以  $tag$  为 0 或 1 来区分尾指针和头指针值相同时对队列的状态是“空”还是“满”。试编写相应的入队和出队的算法。

解: 这就是解决队满队空的三种办法之① 设置一个布尔变量以区别队满还是队空 (其他两种见简答题);

思路: 一开始队空, 设  $tag=0$ , 若从  $rear$  一端加到与  $front$  指针相同时, 表示入队已满, 则令  $tag=1$ ;

若从  $front$  一端加到与  $rear$  指针相同时, 则令  $tag=0$ , 表示出队已空。

3. 试写一个算法判别读入的一个以 '@' 为结束符的字符序列是否是“回文”。

答: 编程如下:

```
int Palindrome_Test() // 判别输入的字符串是否回文序列, 是则返回 1, 否则返回 0
{
 InitStack(S); InitQueue(Q);
 while((c=getchar())!='@')
 {
 Push(S, c); EnQueue(Q, c); // 同时使用栈和队列两种结构
 }
 while(!StackEmpty(S))
 {
 Pop(S, a); DeQueue(Q, b);
 if(a!=b) return ERROR;
 }
 return OK;
} // Palindrome_Test
```

1. 在一个单链表HL中，若要向表头插入一个由指针p指向的结点，则执行( )。
- A. HL=ps p->next=HL  
 B. p->next=HL; HL=p;  
 C. p->next=HL; p=HL;  
 D. p->next=HL->next; HL->next=p;
2. n个顶点的强连通图中至少含有( )。
- A. n-1条有向边  
 B. n条有向边  
 C. n(n-1)/2条有向边  
 D. n(n-1)条有向边
3. 从一棵二叉搜索树中查找一个元素时，其时间复杂度大致为( )。
- A. O(1)  
 B. O(n)  
 C. O(log<sub>2</sub>n)  
 D. O(n<sup>2</sup>)
4. 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
5. 当一个作为实际传递的对象占用的存储空间较大并可能需要修改时，应最好把它说明为( )参数，以节省参数值的传输时间和存储参数的空间。
- A. 整形  
 B. 引用型  
 C. 指针型  
 D. 常值引用型
6. 向一个长度为n的顺序表中插入一个新元素的平均时间复杂度为( )。
- A. O(n)  
 B. O(1)  
 C. O(n<sup>2</sup>)  
 D. O(log<sub>2</sub>n)

- 二、填空题(每小题2分，共28分)
1. 数据的存储结构分为线性和非线性两种。线性结构是指数据元素之间存在着一对一的线性关系。非线性结构是指数据元素之间存在着一对多的非线性关系。
2. 在广义表的存储结构中，单元素结点与表元素结点有一个对应不同，各自分别为原子和子表。
3. 一棵高度为h的B叉树中，最多有 $B^h$ 个叶子结点。
4. 在一棵高度为h的B叉树中，最多有 $B^h$ 个叶子结点。
5. 假定一棵二叉树的结点数为18，则它的最小高度为4，最大高度为18。
6. 在一棵二叉搜索树中，每个分支结点的左子树上所有结点的值一定小于该结点的值，右子树上所有结点的值一定大于该结点的值。
7. 向一个栈顶插入一个具有最小值的元素时，该元素需要向上比较，直到被调整到栈底为止。
8. 表示图的三种存储结构为邻接矩阵、邻接表和十字链表。
9. 对用邻接矩阵表示的具有n个顶点和e条边的图进行任一种遍历，其时间复杂度为O(n<sup>2</sup>)。对用邻接表表示的图进行任一种遍历时，其时间复杂度为O(n+e)。
10. 从有序表(12, 18, 30, 43, 56, 78, 92)中依次二分查找43和56元素时，其查找长度分别为4和5。
11. 假定对长度为n的线性表进行索引顺序查找，并假定每个子表的长度均为1/2。则进行索引顺序查找的平均查找长度为 $\frac{n+1}{2}$ ，时间复杂度为 $O(n)$ 。
12. 一棵B叉树的初始叶子结点数为10。
13. 每次从有序表中取出一个元素，把这个元素插入到有序表中的适当位置，此种排序方法叫插入排序；每次从元素表中取出一个最小或最大元素，把它交换到有序表的一端，此种排序方法叫选择排序。
14. 快速排序在平均情况下的时间复杂度为 $O(n \log_2 n)$ ，最坏情况下的时间复杂度为 $O(n^2)$ 。

三、运算题(每小题6分，共24分)

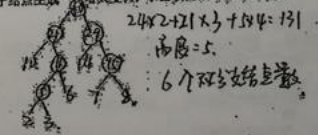
1. 假定一棵二叉树T表示为a(b(c, d), c(((g, s))), 分别写出对它进行先序、中序、后序和后序遍历的结果。
- 先序: a b c d g s  
 中序: b c d g s a  
 后序: c d g s a b
2. 已知一个带权图的顶点集V和边集E分别为:
- V={0, 1, 2, 3, 4, 5};  
 E={(0,1)8, (0,4)5, (0,5)2, (1,4)6, (2,3)25, (2,4)13, (3,5)9, (4,5)10}.
- 则求出该图的最小生成树的权31。
- 最小生成树的权: 31
3. 假定一组记录的排序码为(46, 79, 56, 39, 40, 84, 50, 42)，则利用堆排序方法建立的初始堆为84, 79, 56, 42, 40, 50, 39。
4. 有7个带权结点，其权值分别为3, 7, 8, 2, 6, 10, 14，试以它们为叶子结点生成一棵哈夫曼树，求出该树的带权路径长度、高度、双分支结点数。
- 带权路径长度: 111  
 高度: 5  
 双分支结点数: 6

四、阅读算法，回答问题(每小题8分，共16分)

```

1. VOID AC(List&L)
{
 InitList(L);
 InsertRear(L, 25);
 InsertFront(L, 50);
 Init(A)={5, 8, 12, 15, 36};
 for(int i=0; i<5; i++)
 if (a[i]*2==0) InsertFront(L, a[i]);
 }

```



```

else insertRear(L, a[i]);
}

```

该算法调用执行后，得到的线性表 L 为：36, 12, 2, 5, 15, 5, 15.

```

2. void AG(Queue&Q)
{

```

```

InitQueue(Q);
inta[5] = {6, 12, 5, 15, 8};
for(int i=0; i<5; i++) Qinsert(Q, a[i]);
Qinsert(Q, QDelete(Q));
Qinsert(Q, 20);
Qinsert(Q, QDelete(Q)+18);
while(!QueueEmpty(Q)) cout << QDelete(Q) << " ";
}

```

该算法调用后得到的输出结果为：5, 15, 8, 20, 28.

五、算法填空。在画有横线的地方填写合适的内容(每小题6分，共12分)

1. 从一维数组 A[1..n] 中二分查找关键字为 K 的元素的递归算法，若查找成功则返回对应元素的下标，否则返回 -1。

```

int Binsch(ElemType A[], int low, int high, KeyType K)
{
if (low == high)
{
int mid = (low+high)/2;
if (K == A[mid]) return mid;
else if (K < A[mid]) return Binsch(A, low, mid-1, K);
else return Binsch(A, mid+1, high, K);
}
else return -1;
}

```

2. 已知二叉树中的结点类型 BintreeNode 定义为：

```

struct BintreeNode { BintreeType data; BintreeNode *left, *right;

```

其中 data 为结点数据，left 和 right 分别为指向左、右子女结点的指针。下面函数的功能是返回二叉树 BT 中值为 x 的结点所在的层号，请在画有横线的地方填写合适内容。

```

int NodeLevel(BintreeNode *BT, ElemType X)
{

```

```

if (BT == NULL) return 0; // 空树的层号为 0
else if (BT->data == X) return 1; // 根结点的层号为 1
// 向子树中查找 x 结点
else
{
int c1 = NodeLevel(BT->left, X);
if (c1 != -1) return c1+1;
int c2 = NodeLevel(BT->right, X);
if (c2 != -1) return c2+1;
// 若树中不存在 x 结点则返回 0
else return 0;
}
}

```

六、编写算法(8分)

按所给函数声明编写一个算法，从表头指针为 HL 的单链表中查找出具有最大值的结点，该最大值由函数返回。若单链表为空则中止运行。

```

ElemType MaxValue(LNode*HL);

```

```

{
if (!HL) return -1;
p = HL->next;
max = p->data;
while (p)
{
if (max < p->data) max = p->data;
p = p->next;
}
return max;
}

```

一、单选题(每小题2分，共12分)

评分标准：选对者得2分，否则不得分。

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

二、填空题(每空1分，共20分)

1. 顺序存储 链式存储 索引存储 散列存储(次序无先后)

2. 值(或data) 子表指针(或sublist)

3. 3 x 2, 4 5 / 8

4. 值 子表

5. 5 18

6. 小于 大于(或大于等于)

7. 向上 堆顶

8. 每按矩阵 邻接表 边集数组(次序无先后)

9.  $O(n^2)$   $O(n)$   
10. 1 3

11. 13  $O(n)$

12. 第二层

13. 插入 选择

14. O(nlogn)  $O(n)$

三、填空题(每小题6分,共24分)

1. 顺序: a, b, c, d, e, f, e // 2分

中序: c, b, d, a, f, b, e // 2分

后序: c, d, b, e, f, e, a // 2分

2. 最小生成树的权: 31 // 6分

3. C34, 79, 56, 42, 40, 45, 50, 30 // 6分

4. 最优路径长度: 131 // 3分

高度: 5 // 2分

二分查找次数: 6 // 1分

四、阅读算法,回答问题(每小题8分,共16分)

评分标准: 每小题正确得8分,出现一处错误扣4分,两处及以上错误不得分。

1. C8, 12, 8, 50, 25, 5, 15

2. 5 15 8 6 20 28

五、算法填空,在画有横线的地方填写合适的内容(每小题6分,共12分)

1. return mid // 2分

returnInSch(A, low, mid-1, K) // 2分

returnInSch(A, mid+1, high, K) // 2分

2. NodeLevel(BT->right, X) // 3分

(c>=1)?returnc2+1 // 3分

六、编写算法(8分)

评分标准: 请参考语句后的注释,或根据情况酌情给分。

ElemType MaxValue(LNodeD\* HL.)

{

if (HL==NULL) { // 2分

cerr<<"Linked list is empty!" <<endl;

exit(1);

}

ElemType\* HL->data; // 3分

LNode\* H1->next; // 4分

while(H1) { // 7分

if(max<H1->data)max=H1->data;

H1=H1->next;

}

returnmax; // 8分

}

#### 数据结构复习资料

#### 一、填空题

1. 数据结构是一门研究非数值计算的程序设计问题中计算机的 操作对象 以及它们之间的 关系 和 运算 等的学科。
2. 数据结构被形式地定义为  $(D, R)$ , 其中  $D$  是 数据元素 的有限集合,  $R$  是  $D$  上的 关系 有限集合。
3. 数据结构包括数据的 逻辑结构、数据的 存储结构 和数据的 运算 这三个方面的内容。
4. 数据结构按逻辑结构可分为两大类, 它们分别是 线性结构 和 非线性结构。
5. 线性结构中元素之间存在 一对一 关系, 树形结构中元素之间存在 一对多 关系, 图形结构中元素之间存在 多对多 关系。
6. 在线性结构中, 第一个结点 没有 前驱结点, 其余每个结点有且只有 1 个前驱结点; 最后一个结点 没有 后继结点, 其余每个结点有且只有 1 个后继结点。
7. 在树形结构中, 树根结点没有 前驱 结点, 其余每个结点有且只有 1 个前驱结点; 叶子结点没有 后继 结点, 其余每个结点的后继结点数可以任意多个。
8. 在图形结构中, 每个结点的前驱结点数和后继结点数可以 任意多个。
9. 数据的存储结构可用四种基本的存储方法表示, 它们分别是 顺序、链式、索引 和 散列。

10. 数据的运算最常用的有5种, 它们分别是插入、删除、修改、查找、排序。
11. 一个算法的效率可分为时间效率和空间效率。
12. 在顺序表中插入或删除一个元素, 需要平均移动表中一半元素, 具体移动的元素个数与表长和该元素在表中的位置有关。
13. 线性表中结点的集合是有理数, 结点间的关系是一对一的。
14. 向一个长度为n的向量的第i个元素(1 ≤ i ≤ n+1)之前插入一个元素时, 需向后移动 n-i+1 个元素。
15. 向一个长度为n的向量中删除第i个元素(1 ≤ i ≤ n)时, 需向前移动 n-i 个元素。
16. 在顺序表中访问任意一结点的时间复杂度均为 O(1), 因此, 顺序表也称为随机存取的数据结构。
17. 顺序表中逻辑上相邻的元素的物理位置必定相邻。单链表中逻辑上相邻的元素的物理位置不一定相邻。
18. 在单链表中, 除了首元结点外, 任一结点的存储位置由其直接前驱结点的链域的值指示。
19. 在n个结点的单链表中要删除已知结点\*p, 需找到它的直接前驱结点的地址, 其时间复杂度为 O(n)。
20. 向量、栈和队列都是线性结构, 可以在向量的任何位置插入和删除元素; 对于栈只能在找顶插入和删除元素; 对于队列只能在队尾插入和队首删除元素。
21. 栈是一种特殊的线性表, 允许插入和删除运算的一端称为找顶, 不允许插入和删除运算的一端称为找底。
22. 队列是限定为只能在表的一端进行插入运算, 在表的另一端进行删除运算的线性表。
23. 不包含任何字符(长度为0)的串称为空串; 由一个或多个字符(仅由空格符)组成的串称为空白串。
24. 子串的定位运算称为串的模式匹配; 被匹配的串称为目标串, 子串称为模式。
25. 假设有序线性表 A<sub>1, A<sub>2</sub>, ..., A<sub>n</sub></sub>, 每个元素用相邻的6个字节存放, 存储按字节连续。已知 A<sub>1</sub> 的起始存储位置(基地址)为 1000, 则数组 A 的存储(存储量)为 288B; 元素 A<sub>100</sub> 的第一个字节地址为 1882; 若按行存储时, 元素 A<sub>10</sub> 的第一个字节地址为 (8+4) × 6 + 1600 = 1072; 若按列存储时, 元素 A<sub>10</sub> 的第一个字节地址为 (6 × 7 + 6 + 1000) = 1276。
26. 由3个结点所构成的二叉树有 5 种形态。
27. 一棵深度为6的满二叉树有 n<sub>1</sub>+n<sub>2</sub>+...+n<sub>6</sub>=n-1=31 个分支结点和 2<sup>6</sup>=32 个叶子。  
注: 满二叉树没有度为1的结点, 所以分支结点数就是二度结点数。
28. 一棵具有 2<sup>5</sup>-1 个结点的完全二叉树, 它的深度为 5。  
(注: 用  $\lfloor \log_2(n) \rfloor + 1 = \lfloor 8.32 \rfloor + 1 = 9$ )
29. 设一棵完全二叉树有 700 个结点, 则共有 350 个叶子结点。  
答: 最快方法: 用叶子数 =  $\lfloor n/2 \rfloor = 350$
30. 设一棵完全二叉树具有 1000 个结点, 则此完全二叉树有 500 个叶子结点, 有 499 个度为 2 的结点, 有 1 个结点只有非空左子树, 有 0 个结点只有非空右子树。  
答: 最快方法: 用叶子数 =  $\lfloor n/2 \rfloor = 500$ ,  $n_2 = n - 1 = 499$ 。另外, 最后一结点为 21 属于左叶子, 右叶子是空的, 所以有 1 个非空左子树。完全二叉树的特点决定不可能有左右子树都非空的情况, 所以非空右子树数 = 0。
31. 本数据的存放无规律而言的线性表中进行检索的最佳方法是顺序查找(线性查找)。  
线性有序表 (a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., a<sub>n</sub>) 是从小到大的排列的, 对一个给定的值 k, 用二分法检索表中与 k 相等的元素, 在查找不成功的情况下, 最多需要检索 B 次, 设有 100 个结点, 用二分法查找时, 最大比较次数是 7。
33. 假设在有序线性表 a[20] 上进行折半查找, 则比较一次查找成功的结点数为 1; 比较两次查找成功的结点数为 2; 比较四次查找成功的结点数为 8; 平均查找长度为 3.7。  
解: 显然, 平均查找长度 =  $O(\log_2 n) < 5$  次 (2<sup>5</sup>), 但具体是多少次, 则不应当按照公式  $ASL = \frac{n+1}{n} \log_2(n+1)$  来计算 (即  $(21 \times \log_2 21) / 20 = 4.6$  次并不正确), 因为这是在假设 n=2<sup>n</sup>-1 的情况下推导出来的公式, 应当用列举法罗列:  
全部元素的查找次数为 = (1+2+2+4+3+8+4+5+6) = 74; ASL = 74/20 = 3.7 !!!
34. 折半查找有序表 (4, 6, 20, 38, 50, 70, 88, 100), 若查找表中元素 20, 它将与表中元素 28, 6, 12, 20 比较大小。
35. 在各种查找方法中, 平均查找长度与结点个数 n 无关的查找方法是散列查找。
36. 散列法存储的基本思想是由关键字的值决定数据的存储地址。

二、判断正误 (在正确的说法后面打勾, 反之打叉)

- (X) 1. 链表的每个结点中都恰好包含一个指针。  
答: 错误。链表中的结点可含多个指针域, 分别存放多个指针。例如, 双向链表中的结点可以含有两个指针域, 分别存放指向其直接前趋和直接后继结点的指针。

ALBT  
A[10][4]  
[1] \* 8 + [4] \* 4  
A[4][1]  
(6 \* 7 + 4)

- (X) 2. 链表的物理存储结构具有同链表一样的顺序。错，链表的存储结构特点是无序，而链表的示意图有序。
  - (X) 3. 链表的删除算法很简单，因为当删除链中某个结点后，计算机自动地给后续的各个单元向前移动。错，链表的结点不会移动，只是指针内容改变。
  - (X) 4. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。错，混淆了逻辑结构与物理结构，链表也是线性表！且即使是顺序表，也能存放记录型数据。
  - (X) 5. 顺序表结构适宜于进行顺序存取，而链表适宜于进行随机存取。错，正好说反了，顺序表才适合随机存取，链表恰恰适于“解雇摸瓜”。
  - (X) 6. 顺序存储方式的优点是存储密度大，且插入、删除运算效率高。错，前半正确，但后半说法错误，那是链式存储的优点，顺序存储方式插入、删除运算效率较低，在表长为  $n$  的顺序表中，插入和删除一个数据元素，平均需移动表长一半个数的数据元素。
  - (X) 7. 线性表在物理存储空间中也一定是连续的。错，线性表有两种存储方式，顺序存储和链式存储，后者不要求连续存放。
  - (X) 8. 线性表在顺序存储时，逻辑上相邻的元素未必在存储的物理位置次序上相邻。错误，线性表有两种存储方式，在顺序存储时，逻辑上相邻的元素在存储的物理位置次序上也相邻。
  - (X) 9. 顺序存储方式只能用于存储线性结构。错误，顺序存储方式不仅用于存储线性结构，还可以用来存放非线性结构，例如完全二叉树是属于非线性结构，但其最佳存储方式是顺序存储方式。（后一节介绍）
  - (X) 10. 线性表的逻辑顺序与存储顺序总是一致的。错，理由同7，链式存储就无庸一致。
  - (X) 11. 线性表的每个结点只能是一个简单类型，而链表的每个结点可以是一个复杂类型。错，线性表是逻辑结构概念，可以顺序存储或链式存储，与元素数据类型无关。
  - (X) 12. 在表结构中最常用的是线性表，栈和队列不太常用。错，不一定吧？调用于程序或函数常用，CPU 中也用队列。
  - (√) 13. 栈是一种对所有插入、删除操作限于在表的一端进行的线性表，是一种后进先出型结构。
  - (√) 14. 对于不同的使用者，一个表结构既可以是栈，也可以是队列，也可以是线性表。正确，都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。
  - (X) 15. 栈和链表是两种不同的数据结构。错，栈是逻辑结构的名称，是特殊线性表，而链表是存储结构概念，二者不是同类项。
  - (X) 16. 栈和队列是一种非线性数据结构。错，他们都是线性逻辑结构，栈和队列其实是特殊的线性表，对运算的定义略有不同而已。
  - (√) 17. 栈和队列的存储方式既可是顺序方式，也可是链接方式。
  - (√) 18. 两个栈共享一片连续内存空间时，为提高内存利用率，减少溢出机会，应把两个栈的栈底分别设在这片内存空间的两端。
  - (X) 19. 队是一种插入与删除操作分别在表的两端进行的线性表，是一种先进后出型结构。错，后半句不对。
  - (X) 20. 一个栈的输入序列是 12345，则栈的输出序列不可能是 12345。错，有可能。
  - (√) 21. 若二叉树用二叉链表作存储结构，则在  $n$  个结点的二叉树链表中只有  $n-1$  个非空指针域。
  - (X) 22. 二叉树中每个结点的两棵子树的高度差等于 1。
  - (√) 23. 二叉树中每个结点的两棵子树是有序的。
  - (X) 24. 二叉树中每个结点有两棵非空子树或有两棵空子树。
  - (X) 25. 二叉树中每个结点的关键字值大于其左非空子树（若存在的话）所有结点的关键字值，且小于其右非空子树（若存在的话）所有结点的关键字值。（应当是二叉排序树的特点）
  - (X) 26. 二叉树中所有结点的个数是  $2^k-1$ ，其中  $k$  是树的深度。（应  $2^{k+1}-1$ ）
  - (X) 27. 二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。
  - (X) 28. 对于一棵非空二叉树，它的根结点作为第一层，则它的第  $i$  层上最多能有  $2^{i-1}$  个结点。（应  $2^i$ ）
  - (√) 29. 用二叉链表法 (link-rlink) 存储包含  $n$  个结点的二叉树，结点的  $2n$  个指针区域中有  $n+1$  个为空指针。
  - (√) 30. 具有 12 个结点的完全二叉树有 5 个度为 2 的结点。
- 三、单项选择题
- B) 1. 非线性结构是数据元素之间存在一种：

- A) 一对多关系 B) 多对多关系 C) 多对一关系 D) 一对一关系
- (C) 2. 数据结构中, 与所使用的计算机无关的是数据的\_\_\_\_\_结构:  
A) 存储 B) 物理 C) 逻辑 D) 物理和存储
- (C) 3. 算法分析的目的是:  
A) 找出数据结构的合理性 B) 研究算法中的输入和输出的关系  
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
- (A) 4. 算法分析的两个主要方面是:  
A) 空间复杂性和时间复杂性 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
- (C) 5. 计算机算法指的是:  
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
- (B) 6. 计算机算法必须具备输入、输出和\_\_\_\_\_等5个特性。  
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
- (C) 7. 数据在计算机存储器内表示时, 物理地址与逻辑地址相同并且是连续的, 称之为:  
(A) 存储结构 (B) 逻辑结构 (C) 顺序存储结构 (D) 链式存储结构
- (B) 8. 一个向量第一个元素的存储地址是100, 每个元素的长度为2, 则第5个元素的地址是\_\_\_\_\_  
(A) 110 (B) 108 (C) 100 (D) 120
- (A) 9. 在n个结点的顺序表中, 算法的时间复杂度是O(1)的操作是:  
(A) 访问第i个结点 ( $1 \leq i \leq n$ ) 和求第i个结点的直接前驱 ( $2 \leq i \leq n$ )  
(B) 在第i个结点后插入一个新结点 ( $1 \leq i \leq n$ )  
(C) 删除第i个结点 ( $1 \leq i \leq n$ )  
(D) 将n个结点从小到大排序
- (B) 10. 向一个含有127个元素的顺序表中插入一个新元素并保持原来顺序不变, 平均要移动\_\_\_\_\_个元素  
(A) 8 (B) 63.5 (C) 63 (D) 7
- (A) 11. 链接存储的存储结构所占存储空间:  
(A) 分两部分, 一部分存放结点值, 另一部分存放表示结点间关系的指针  
(B) 只有一部分, 存放结点值  
(C) 只有一部分, 存放表示结点间关系的指针  
(D) 分两部分, 一部分存放结点值, 另一部分存放结点所占单元数
- (B) 12. 链表是一种采用\_\_\_\_\_存储结构存储的线性表:  
(A) 顺序 (B) 链式 (C) 星式 (D) 网状
- (D) 13. 线性表若采用链式存储结构时, 要求内存中可用存储单元的地址:  
(A) 必须是连续的 (B) 部分地址必须是连续的  
(C) 一定是不连续的 (D) 连续或不连续都可以
- (B) 14. 线性表L在\_\_\_\_\_情况下适用于使用链式结构实现。  
(A) 需经常修改L中的结点值 (B) 需不断对L进行删除插入  
(C) L中含有大量的结点 (D) L中结点结构复杂
- (B) 15. 栈中元素的进出原则是  
A. 先进先出 B. 后进先出 C. 栈空则进 D. 栈满则出
- (C) 16. 若已知一个栈的入栈序列是1, 2, 3, ..., n, 其输出序列为p1, p2, p3, ..., pn, 若p1=n, 则pi为  
A. i B. n-i C. n-i+1 D. 不确定
- (B) 17. 判定一个栈ST (栈多元素为m) 为空的条件是  
A. ST-top > 0 B. ST-top = 0 C. ST-top < m D. ST-top = m
- (C) 18. 在一个图中, 所有顶点的度数之和等于图的边数的\_\_\_\_\_倍。  
A. 1/2 B. 1 C. 2 D. 4
- (B) 19. 在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的\_\_\_\_\_倍。  
A. 1/2 B. 1 C. 2 D. 4



20. 有 8 个结点的无向图最多有          条边。  
 A. 14                      B. 28                      C. 56                      D. 112
21. 有 8 个结点的有向完全图有          条边。  
 A. 14                      B. 28                      C. 56                      D. 112
22. 在表长为  $n$  的链表中进行线性查找，它的平均查找长度为  
 A.  $ASL = n$ ;                      B.  $ASL = (n+1) / 2$ ;  
 C.  $ASL = \sqrt{n+1}$ ;                      D.  $ASL \approx 1.09 \lg_2(n+1) - 1$
23. 折半查找有序表 (4, 6, 10, 12, 20, 30, 50, 70, 88, 100)，若查找表中元素 56，则它将依次与表中          比较大小，查找结果是失败。  
 A. 20, 70, 30, 50                      B. 30, 88, 70, 50                      C. 20, 50                      D. 30, 88, 50
24. 对 22 个记录的有序表作折半查找，当查找失败时，至少需要比较          次关键字。  
 A. 3                      B. 4                      C. 5                      D. 6
25. 链表适用于          查找。  
 A. 顺序                      B. 二分法                      C. 顺序，也能二分法                      D. 随机

《数据结构与算法》复习题

- 一、选择题。
1. 在数据结构中，从逻辑上可以把数据结构分为 C。  
 A. 动态结构和静态结构                      B. 紧凑结构和非紧凑结构  
 C. 线性结构和非线性结构                      D. 内部结构和外部结构
2. 数据结构在计算机内存中的表示是指 A。  
 A. 数据的存储结构                      B. 数据逻辑结构                      C. 数据的逻辑结构                      D. 数据元素之间的关系
3. 在数据结构中，与所使用的计算机无关的是数据的 A 结构。  
 A. 逻辑                      B. 存储                      C. 逻辑和存储                      D. 物理
4. 在存储数据时，通常不仅要存储各数据元素的值，而且还要存储 C。  
 A. 数据的处理方法                      B. 数据元素的类型  
 C. 数据元素之间的关系                      D. 数据的存储方法
5. 在决定选取何种存储结构时，一般不考虑 A。  
 A. 各结点的值如何                      B. 结点个数的多少  
 C. 对数据有哪些运算                      D. 所用的编程语言实现这种结构是否方便。
6. 以下说法正确的是 D。  
 A. 数据项是数据的基本单位  
 B. 数据元素是数据的最小单位  
 C. 数据结构是带结构的数据项的集合  
 D. 一些表面上很不相同的数据可以有相同的逻辑结构
7. 算法分析的目的是 C，算法分析的两个主要方面是 A。  
 (1) A. 找出数据结构的合理性                      B. 研究算法中的输入和输出的关系  
 C. 分析算法的效率以求改进                      D. 分析算法的可读性和文档性  
 (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
8. 下面程序段的时间复杂度是  $O(n^2)$ 。  
 $s = 0$ ;  
 for (  $i = 0; i < n; i++$  )  
 for (  $j = 0; j < n; j++$  )  
 $s += b[i][j]$ ;

9. 下面程序段的时间复杂度是  $O(n^2)$ 。

```
for (i = 0; i < n; i++)
 for (j = 0; j < m; j++)
 A[i][j] = 0;
```

10. 下面程序段的时间复杂度是  $O(\log n)$ 。

```
i = 0;
while (i <= n)
 i = i * 3;
```

11. 在以下的叙述中，正确的是 B。

- A. 线性表的顺序存储结构优于链表存储结构
- B. 二维数组是其数据元素为线性表的线性表
- C. 栈的操作方式是先进先出
- D. 队列的操作方式是先进后出

12. 通常要求同一逻辑结构中的所有数据元素具有相同的特性，这意味着 B。

- A. 数据元素具有同一特点
- B. 不仅数据元素所包含的数据项的个数要相同，而且对应的数据项的类型要一致
- C. 每个数据元素都一样
- D. 数据元素所包含的数据项的个数要相等

13. 链表不具备的特点是 A。

- A. 可随机访问任一结点
- B. 插入删除不需要移动元素
- C. 不必事先估计存储空间
- D. 所需空间与其长度成正比

14. 不带头结点的单链表 head 为空的判定条件是 A。

- A. head == NULL
- B. head->next == NULL
- C. head->next == head
- D. head != NULL

15. 带头结点的单链表 head 为空的判定条件是 B。

- A. head == NULL
- B. head->next == NULL
- C. head->next == head
- D. head != NULL

16. 若某表最常用的操作是在最后一个结点之后插入一个结点或删除最后一个结点，则采用 D 存储方式最节省运算时间。

- A. 单链表
- B. 给出表头指针的循环链表
- C. 双链表
- D. 带头结点的双循环链表

17. 若要节省存储空间，插入和删除不需要移动元素的线性表，其存储结构是 B。

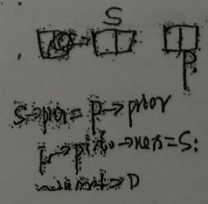
- A. 单链表
- B. 静态链表
- C. 线性链表
- D. 顺序存储结构

18. 非空的循环单链表 head 的尾结点 (由 p 所指向) 满足 C。

- A. p->next == NULL
- B. p == NULL
- C. p->next == head
- D. p == head

19. 在循环双链表的 p 所指的结点之前插入 s 所指结点的操作是 D。

- A. p->prior = s; s->next = p; p->prior->next = s; s->prior = p->prior
- B. p->prior = s; p->prior->next = s; s->next = p; s->prior = p->prior



C.  $s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior; p \rightarrow prior = s; p \rightarrow prior \rightarrow next = s$   
 D.  $s \rightarrow next = p; s \rightarrow prior = p \rightarrow prior; p \rightarrow prior \rightarrow next = s; p \rightarrow prior = s$

20. 如果最常用的操作是取第  $i$  个结点及其前驱, 则采用 D 存储方式最节省时间。  
 A. 单链表 B. 双链表 C. 单循环链表 D. 顺序表

21. 在一个具有  $n$  个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是 B。  
 A.  $O(1)$  B.  $O(n)$  C.  $O(n^2)$  D.  $O(n \log_2 n)$

22. 在一个长度为  $n$  ( $n > 1$ ) 的单链表上, 设有头和尾两个指针, 执行 B 操作与链表的长度有关。  
 A. 删除单链表中的第一个元素  
 B. 删除单链表中的最后一个元素  
 C. 在单链表第一个元素前插入一个新元素  
 D. 在单链表最后一个元素后插入一个新元素

23. 与单链表相比, 双链表的优点之一是 D。  
 A. 插入、删除操作更简单  
 B. 可以进行随机访问  
 C. 可以省略表头指针或表尾指针  
 D. 顺序访问相邻结点更灵活

24. 如果对线性表的操作只有两种, 即删除第一个元素, 在最后一个元素的后面插入新元素, 则最好使用 B。  
 A. 只有表头指针没有表尾指针的循环单链表  
 B. 只有表尾指针没有表头指针的循环单链表  
 C. 非循环双链表  
 D. 循环双链表

25. 在长度为  $n$  的顺序表的第  $i$  个位置上插入一个元素 ( $1 \leq i \leq n+1$ ), 元素的移动次数为: A。  
 A.  $n - i + 1$  B.  $n - i$  C.  $i$  D.  $i - 1$

26. 对于只在表的首、尾两端进行插入操作的线性表, 宜采用的存储结构为 C。  
 A. 顺序表 B. 用头指针表示的循环单链表  
 C. 用尾指针表示的循环单链表 D. 单链表

27. 下述哪一条是顺序存储结构的优点? C。  
 A. 插入运算方便 B. 可方便地用于各种逻辑结构的存储表示  
 C. 存储密度大 D. 删除运算方便

28. 下面关于线性表的叙述中, 错误的是哪一个? B。  
 A. 线性表采用顺序存储, 必须占用一片连续的存储单元  
 B. 线性表采用顺序存储, 便于进行插入和删除操作。  
 C. 线性表采用链式存储, 不必占用一片连续的存储单元  
 D. 线性表采用链式存储, 便于进行插入和删除操作。

29. 线性表是具有  $n$  个 B 的有限序列。  
 A. 字符 B. 数据元素 C. 数据项 D. 表元素

30. 在  $n$  个结点的线性表的数组实现中, 算法的时间复杂度是  $O(1)$  的操作是 A。  
 A. 访问第  $i$  ( $1 \leq i \leq n$ ) 个结点和求第  $i$  个结点的直接前驱 ( $1 < i \leq n$ )  
 B. 在第  $i$  ( $1 \leq i \leq n$ ) 个结点后插入一个新结点

- C. 删除第  $i$  ( $1 < i < n$ ) 个结点  
D. 以上都不对
31. 若长度为  $n$  的线性表采用顺序存储结构, 在其第  $i$  个位置插入一个新元素的算法的时间复杂度为 C。  
A.  $O(0)$  B.  $O(1)$  C.  $O(n)$  D.  $O(n^2)$
32. 对于顺序存储的线性表, 访问结点和增加、删除结点的时间复杂度为 C。  
A.  $O(n)$   $O(n)$  B.  $O(n)$   $O(1)$  C.  $O(1)$   $O(n)$  D.  $O(1)$   $O(1)$
33. 线性表  $(a_1, a_2, \dots, a_n)$  以链式方式存储, 访问第  $i$  位置元素的时间复杂度为 C。  
A.  $O(0)$  B.  $O(1)$  C.  $O(n)$  D.  $O(n^2)$
34. 单链表中, 增加一个头结点的目的是为了 C。  
A. 使单链表至少有一个结点 B. 探识表结点中首结点的位置  
C. 方便运算的实现 D. 说明单链表是线性表的链式存储
35. 在单链表指针为  $p$  的结点之后插入指针为  $s$  的结点, 正确的操作是 B。  
A.  $p \rightarrow \text{next} = s; s \rightarrow \text{next} = p \rightarrow \text{next}$  B.  $s \rightarrow \text{next} = p \rightarrow \text{next}; p \rightarrow \text{next} = s$   
C.  $p \rightarrow \text{next} = s; p \rightarrow \text{next} = s \rightarrow \text{next}$  D.  $p \rightarrow \text{next} = s \rightarrow \text{next}; p \rightarrow \text{next} = s$
36. 线性表的顺序存储结构是一种 A。  
A. 随机存取的结构 B. 顺序存取的结构  
C. 索引存取的结构 D. Hash 存取的结构
37. 栈的特点是 B, 队列的特点是 A。  
A. 先进先出 B. 先进后出
38. 栈和队列的共同点是 C。  
A. 都是先进后出 B. 都是先进先出  
C. 只允许在端点处插入和删除元素 D. 没有共同点
39. 一个栈的进栈序列是  $a, b, c, d, e$ , 则栈的不可能的输出序列是 C。  
A.  $edcba$  B.  $decba$  C.  $dceab$  D.  $ebcde$
40. 设有一个栈, 元素依次进栈的顺序为  $A, B, C, D, E$ 。下列 C 是不可能的出栈序列。  
A.  $A, B, C, D, E$  B.  $B, C, D, E, A$  C.  $E, A, B, C, D$  D.  $E, D, C, B, A$
41. 以下 B 不是队列的基本运算?  
A. 从队尾插入一个新元素 B. 从队列中删除第  $i$  个元素  
C. 判断一个队列是否为空 D. 读取队头元素的值
42. 若已知一个栈的进栈序列是  $1, 2, 3, \dots, n$ , 其输出序列为  $p_1, p_2, p_3, \dots, p_n$ , 若  $p_1 = n$ , 则  $p_i$  为 C。  
A.  $i$  B.  $n-i$  C.  $n-i+1$  D. 不确定
43. 判定一个顺序栈  $st$  (最多元素为  $MaxSize$ ) 为空的条件是 B。  
A.  $st \rightarrow \text{top} != -1$  B.  $st \rightarrow \text{top} == -1$   
C.  $st \rightarrow \text{top} != MaxSize$  D.  $st \rightarrow \text{top} == MaxSize$
44. 判定一个顺序栈  $st$  (最多元素为  $MaxSize$ ) 为满的条件是 D。

- A.  $st \rightarrow top \neq -1$
- B.  $st \rightarrow top == -1$
- C.  $st \rightarrow top != MaxSize$
- D.  $st \rightarrow top == MaxSize$

45. 一个队列的入队序列是 1, 2, 3, 4, 则队列的输出序列是 B。  
 A. 4, 3, 2, 1      B. 1, 2, 3, 4  
 C. 1, 4, 3, 2      D. 3, 2, 4, 1

46. 判定一个循环队列  $qu$  (最多元素为  $MaxSize$ ) 为空的条件是 C。  
 A.  $qu \rightarrow rear - qu \rightarrow front == MaxSize$       B.  $qu \rightarrow rear - qu \rightarrow front - 1 == MaxSize$   
 C.  $qu \rightarrow rear == qu \rightarrow front$       D.  $qu \rightarrow rear == qu \rightarrow front - 1$

47. 在循环队列中, 若  $front$  与  $rear$  分别表示队头元素和队尾元素的位置, 则判断循环队列空的条件是 C。  
 A.  $front == rear + 1$       B.  $rear == front + 1$       C.  $front == rear$       D.  $front == 0$

48. 向一个栈顶指针为  $h$  的带头结点的链栈中插入指针  $s$  所指的结点时, 应执行 D 操作。  
 A.  $h \rightarrow next = s$ ;      B.  $s \rightarrow next = h$ ;  
 C.  $s \rightarrow next = h$ ;  $h = s$ ;      D.  $s \rightarrow next = h \rightarrow next$ ;  $h \rightarrow next = s$ ;

49. 输入序列为 ABC, 可以变为 CBA 时, 经过的栈操作为 B。  
 A. push, pop, push, pop, push, pop      B. push, push, push, pop, pop, pop  
 C. push, push, pop, pop, push, pop      D. push, pop, push, push, pop, pop

50. 若栈采用顺序存储方式存储, 现两栈共享空间  $V[1..m]$ ,  $top[1]$ 、 $top[2]$  分别代表第 1 和第 2 个栈的栈顶, 栈 1 的底在  $V[1]$ , 栈 2 的底在  $V[m]$ , 则栈满的条件是 B。  
 A.  $|top[2] - top[1]| = 0$       B.  $top[1] + 1 = top[2]$       C.  $top[1] + top[2] = m$       D.  $top[1] = top[2]$

51. 设计一个判别表达式中左、右括号是否配对出现的算法, 采用 D 数据结构最佳。  
 A. 线性表的顺序存储结构      B. 队列      C. 线性表的链式存储结构      D. 栈

52. 允许对队列进行的操作有 D。  
 A. 对队列中的元素排序      B. 取出最近进队的元素  
 C. 在队头元素之前插入元素      D. 删除队头元素

53. 对于循环队列 D。  
 A. 无法判断队列是否为空      B. 无法判断队列是否为满  
 C. 队列不可能满      D. 以上说法都不对

54. 若用一个大小为 6 的数组来实现循环队列, 且当前  $rear$  和  $front$  的值分别为 0 和 3, 当从队列中删除一个元素, 再加入两个元素后,  $rear$  和  $front$  的值分别为 B。  
 A. 1 和 5      B. 2 和 4      C. 4 和 2      D. 5 和 1

55. 队列的“先进先出”特性是指 D。  
 A. 最早插入队列中的元素总是最后被删除  
 B. 当同时进行插入、删除操作时, 总是插入操作优先  
 C. 每当有删除操作时, 总是要先做一次插入操作  
 D. 每次从队列中删除的总是最早插入的元素

56. 与顺序栈相比, 链栈有一个比较明显的优势是 A。  
 A. 通常不会出现栈满的情况      B. 通常不会出现栈空的情况

- C. 插入操作更容易实现 D. 删除操作更容易实现

57. 用不带头结点的单链表存储队列, 其头指针指向队头结点, 尾指针指向队尾结点, 则在进行出队操作时 C。  
 A. 仅修改队头指针 B. 仅修改队尾指针  
 C. 队头、队尾指针都要修改 D. 队头、队尾指针都要修改

58. 若串 S="software", 其子串的数目是 B。  
 A. 8 B. 37 C. 36 D. 9  
 $2+2+8 \quad 8+7+6+5+4$   
 $215$

59. 串的长度是指 B。  
 A. 串中所含不同字母的个数 B. 串中所含字符的个数  
 C. 串中所含不同字符的个数 D. 串中所含非空格字符的个数

60. 串是一种特殊的线性表, 其特殊性体现在 B。  
 A. 可以顺序存储 B. 数据元素是一个字符  
 C. 可以链式存储 D. 数据元素可以是多个字符

61. 设有两个串 p 和 q, 求 q 在 p 中首次出现的位置的运算称为 B。  
 A. 连接 B. 模式匹配 C. 求子串 D. 求串长

62. 数组 A 中, 每个元素的长度为 3 个字节, 行下标 i 从 1 到 6, 列下标 j 从 1 到 10, 从首地址 SA 开始连续存放的存储器内, 该数组按行存放, 元素 A[8][5] 的起始地址为 C。  
 A. SA+141 B. SA+144 C. SA+222 D. SA+225  
 $1 \times 8 \times 10 = 80$   
 $8 \times 3 + 5 \times 3 = 39$   
 $80 + 39 = 119$   
 $119 + 3 = 122$

63. 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 开始连续存放的存储器内, 该数组按行存放, 元素 A[8][8] 的起始地址为 C。  
 A. SA+141 B. SA+180 C. SA+222 D. SA+225  
 $1 \times 8 \times 10 = 80$   
 $8 \times 3 + 8 \times 3 = 48$   
 $80 + 48 = 128$   
 $128 + 3 = 131$

64. 若声明一个浮点数组如下: float average[]=new float[30]; 假设该数组的内存起始位置为 200, average[15] 的内存地址是 C。  
 A. 214 B. 215 C. 260 D. 256  
 $50 \times 4 = 200$   
 $15 \times 4 = 60$   
 $200 + 60 = 260$

65. 设二维数组 A[1... m, 1... n] 按行存储在数组 B 中, 则二维数组元素 A[i, j] 在一维数组 B 中的下标为 A。  
 A. n\*(i-1)+j B. n\*(i-1)+j-1 C. i\*(j-1) D. j\*n+i-1  
 $(i-1) \times n + j$

66. 有一个 100x90 的稀疏矩阵, 非 0 元素有 10, 设每个整数占 2 个字节, 则用三元组表示该矩阵时, 所需的字节数是 B。  
 A. 20 B. 66 C. 18 000 D. 33  
 $2 \times 100 \times 90 = 18000$

67. 数组 A[0... 4, -1... -3, 5... 7] 中含有的元素个数是 A。  
 A. 55 B. 45 C. 36 D. 16

68. 对矩阵进行压缩存储是为了 D。  
 A. 方便观察 B. 方便存储 C. 提高运算速度 D. 减少存储空间

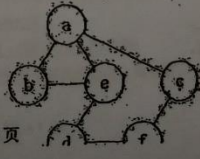
69. 设有一个 10 阶的对称矩阵 A, 采用压缩存储方式, 以行序为主存储, a<sub>11</sub> 为第一个元素, 其存储地址为 1, 每个元素占 1 个地址空间, 则 a<sub>88</sub> 的地址为 B。  
 A. 13 B. 33 C. 18 D. 40

70. 稀疏矩阵一般的压缩存储方式有两种, 即 C。

- 三元组和三元链表      B. 三元组和散列  
 三元组和十字链表      C. 散列和十字链表

71. 树最适合用来表示 C .
- 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
72. 深度为5的二叉树至多有 C 个结点.
- A. 16    B. 32    C. 31    D. 10
73. 对一个满二叉树,  $m$  个叶子,  $n$  个结点, 深度为  $h$ , 则 D .
- A.  $n = h+m$     B.  $h+m = 2n$     C.  $m = h-1$     D.  $n = 2^h - 1$
74. 任何一棵二叉树的叶子结点在前序、中序和后序遍历序列中的相对次序 A .
- A. 不发生改变    B. 发生改变    C. 不能确定    D. 以上都不对
75. 在线索化树中, 每个结点必须设置一个标志来说明它的左、右链指向的是树结构信息, 还是线索化信息, 若0标识树结构信息, 1标识线索, 对应叶结点的左右链域, 应标识为 D .
- A. 00    B. 01    C. 10    D. 11
76. 在下列论述中, 正确的是 D .
- ①只有一个结点的二叉树的度为0; ②二叉树的度为2; ③二叉树的左右子树可任意交换;  
④深度为  $K$  的顺序二叉树的结点个数小于或等于深度相同的满二叉树.
- A. ①②③    B. ②③④    C. ②④    D. ①④
77. 设森林  $F$  对应的二叉树为  $B$ , 它有  $m$  个结点,  $B$  的根为  $p$ ,  $p$  的右子树的结点数为  $n$ , 森林  $F$  中第一棵树的结点的个数是 A .
- A.  $m-n$     B.  $m-n-1$     C.  $n+1$     D. 不能确定
78. 若一棵二叉树具有10个度为2的结点, 5个度为1的结点, 则度为0的结点的个数是 B .
- A. 9    B. 11    C. 15    D. 不能确定
79. 具有10个叶子结点的二叉树中有 B 个度为2的结点.
- A. 8    B. 9    C. 10    D. 11
80. 在一个无向图中, 所有顶点的度数之和等于所有边数的 C 倍.
- A.  $1/2$     B. 1    C. 2    D. 4
81. 在一个有向图中, 所有顶点的入度之和等于所有顶点的出度之和的 B 倍.
- A.  $1/2$     B. 1    C. 2    D. 4
82. 某二叉树结点的中序序列为 ABCDEFG, 后序序列为 BDCAFGE, 则其左子树中结点数目为: C
- A. 3    B. 2    C. 4    D. 5
83. 已知一算术表达式的中缀形式为  $A+B * C - D / E$ , 后缀形式为  $ABC + * DE / -$ , 其前缀形式为 D .
- A.  $-A+B * C / DE$     B.  $-A+B * C D / E$     C.  $-+ABC / DE$     D.  $-+A * BC / DE$

84. 已知一个图, 如图所示, 若从顶点  $a$  出发按深度搜索法进行遍历, 为 D; 按广度搜索法进行遍历, 则可能得到的一种顶点序列为 A.
- ①A.  $a, b, e, c, d, f$     B.  $a, c, f, e, b, d$



则可能得到的一种顶点序列 A

- C. a, e, b, c, f, d.      D. a, e, d, f, c, b  
 ② A. a, b, c, e, d, f      B. a, b, c, e, f, d  
 C. a, e, b, c, f, d.      D. a, c, f, d, e, b
85. 采用邻接表存储的图的深度优先遍历算法类似于二叉树的 A。  
 A. 先序遍历    B. 中序遍历    C. 后序遍历    D. 按层遍历
86. 采用邻接表存储的图的广度优先遍历算法类似于二叉树的 D。  
 A. 先序遍历    B. 中序遍历    C. 后序遍历    D. 按层遍历
87. 具有  $n$  个结点的连通图至少有 A 条边。  
 A.  $n-1$     B.  $n$     C.  $n(n-1)/2$     D.  $2n$
88. 广义表  $((a), a)$  的表头是 C, 表尾是 C。  
 A.  $a$     B.  $()$     C.  $(a)$     D.  $((a))$
89. 广义表  $((a))$  的表头是 C, 表尾是 B。  
 A.  $a$     B.  $()$     C.  $(a)$     D.  $((a))$
90. 顺序查找法适合于存储结构为 B 的线性表。  
 A. 散列存储    B. 顺序存储或链式存储    C. 压缩存储    D. 索引存储
91. 对线性表进行折半查找时, 要求线性表必须 B。  
 A. 以顺序方式存储    B. 以顺序方式存储, 且结点按关键字有序排列  
 C. 以链式方式存储    D. 以链式方式存储, 且结点按关键字有序排列
92. 采用折半查找法查找长度为  $n$  的线性表时, 每个元素的平均查找长度为 D。  
 A.  $O(n)$     B.  $O(\log_2 n)$     C.  $O(n)$     D.  $O(\log_2 n)$
93. 有一个有序表为  $\{1, 3, 9, 12, 32, 41, 45, 62, 75, 77, 82, 95, 100\}$ , 当折半查找值为 82 的结点时, C 次比较后查找成功。  
 A. 11    B. 5    C. 4    D. 8
94. 二叉树为二叉排序树的充分必要条件是任一结点的值均大于其左孩子的值、小于其右孩子的值。这种说法 B。  
 A. 正确    B. 错误
95. 下面关于 B 树和 B+ 树的叙述中, 不正确的结论是 A。  
 A. B 树和 B+ 树都能有效的支持顺序查找    B. B 树和 B+ 树都能有效的支持随机查找  
 C. B 树和 B+ 树都是平衡的多叉树    D. B 树和 B+ 树都可用于文件索引结构
96. 以下说法错误的是 B。  
 A. 散列表存储的思想是由关键字值决定数据的存储地址  
 B. 散列表的结点中只包含数据元素自身的信息, 不包含指针。  
 C. 负载因子是散列表的一个重要参数, 它反映了散列表的饱满程度。  
 D. 散列表的查找效率主要取决于散列表构造时选取的散列函数和处理冲突的方法。
97. 查找效率最高的二叉排序树是 C。  
 A. 所有结点的左子树都为空的二叉排序树。  
 B. 所有结点的右子树都为空的二叉排序树。



...二叉树。  
...有左子树的二叉排序树。

98. 排序方法中,从未排序序列中依次取出元素与已排序序列中的元素进行比较,将其放入已排序序列的正确位置上的方法,称为 C。  
A. 希尔排序 B. 冒泡排序 C. 插入排序 D. 选择排序

99. 在所有的排序方法中,关键字比较的次数与记录的初始排列次序无关的是 D。  
A. 希尔排序 B. 冒泡排序 C. 直接插入排序 D. 直接选择排序

100. 堆是一种有用的数据结构,下列关键码序列 D 是一个堆。  
A. 94, 31, 53, 23, 16, 72 B. 94, 53, 31, 72, 16, 23  
C. 16, 53, 23, 94, 31, 72 D. 16, 31, 23, 94, 53, 72

101. 堆排序是一种 B 排序。  
A. 插入 B. 选择 C. 交换 D. 归并

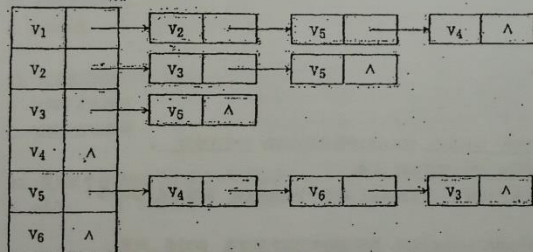
102. D 在链表中进行操作比在顺序表中进行操作效率高。  
A. 顺序查找 B. 折半查找 C. 分块查找 D. 插入

103. 直接选择排序的时间复杂度为 D。(n 为元素个数)  
A.  $O(n)$  B.  $O(\log n)$  C.  $O(n \log n)$  D.  $O(n^2)$

## 二、填空题。

- 数据逻辑结构包括 线性结构、树形结构 和 图状结构 三种类型,树形结构和图状结构合称 非线性结构。
- 数据的逻辑结构分为 集合、线性结构、树形结构 和 图状结构 4 种。
- 在线性结构中,第一个结点 没有 前驱结点,其余每个结点有且只有 1 个前驱结点;最后一个结点 没有 后继结点,其余每个结点有且只有 1 个后继结点。
- 线性结构中元素之间存在 一对一 关系,树形结构中元素之间存在 一对多 关系,图状结构中元素之间存在 多对多 关系。
- 在树形结构中,树根结点没有 前驱 结点,其余每个结点有且只有 1 个前驱结点;叶子结点没有 后继 结点,其余每个结点的后继结点可以 任意多个。
- 数据结构的基本存储方法是 顺序、链式、索引 和 散列 存储。
- 衡量一个算法的优劣主要考虑正确性、可读性、健壮性和 时间复杂度 与 空间复杂度。
- 评估一个算法的优劣,通常从 时间复杂度 和 空间复杂度 两个方面考察。
- 算法的 5 个重要特性是 有穷性、确定性、可行性、输入和输出。
- 在一个长度为 n 的顺序表中删除第 i 个元素时,需向前移动  $n-i-1$  个元素。
- 在单链表中,要删除某一指定的结点,必须找到该结点的 前驱 结点。
- 在双链表中,每个结点有两个指针域,一个指向 前驱 结点,另一个指向 后继 结点。
- 在顺序表中插入或删除一个数据元素,需要平均移动 n 个数据元素,移动数据元素的个数与 位置 有关。
- 当线性表的元素总数基本稳定,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的元素是,应采用 顺序 存储结构。
- 根据线性表的链式存储结构中每一个结点包含的指针个数,将线性链表分成 单链表 和 双链表。
- 顺序存储结构是通过 下标 表示元素之间的关系的;链式存储结构是通过 指针 表示元素之间的关系的。
- 带头结点的循环链表 L 中只有一个元素结点的条件是  $L \rightarrow \text{next} \rightarrow \text{next} = L$ 。
- 栈 是限定仅在表尾进行插入或删除操作的线性表,其运算遵循 后进先出 的原则。
- 空串是 零个字符的串,其长度等于 零,空白串是由 一个或多个空格字符组成的串,其长度等于 其包含的空格个数。
- 组成串的数据元素只能是 单个字符。

21. 一个字符串中 任意个连续字符构成的部分 称为该串的子串。
22. 字符串 "str" 在主串 "datastructure" 中的位置是 5。
23. 二维数组 M 的每个元素是 6 个字符组成的串，行下标 i 的范围从 0 到 8，列下标 j 的范围从 1 到 10，则存放 M 至少需要 546 个字节；M 的第 8 列和第 5 行共占 108 个字节。
24. 稀疏矩阵一般的压缩存储方法有两种，即 三元组表 和 十字链表。
25. 广义表 ((a), ((b), c), (((d)))) 的长度是 3，深度是 4。
26. 在一棵二叉树中，度为零的结点的个数为  $n_0$ ，度为 2 的结点的个数为  $n_2$ ，则有  $n_0 =$   $n_2 + 1$ 。
27. 在有  $n$  个结点的二叉链表中，空链域的个数为  $n + 1$ 。
28. 一棵有  $n$  个叶子结点的哈夫曼树共有  $2n - 1$  个结点。
29. 深度为 5 的二叉树至多有 31 个结点。
30. 若某二叉树有 20 个叶子结点，有 30 个结点仅有一个孩子，则该二叉树的总结点数为 69。
31. 某二叉树的前序遍历序列是 abdgcefh，中序序列是 dghaecfh，其后序序列为 gdbehfca。
32. 线索二叉树的左线索指向其 遍历序列中的前驱，右线索指向其 遍历序列中的后继。
33. 在各种查找方法中，平均查找长度与结点个数  $n$  无关的查找方法是 散列查找法。
34. 在分块索引查找方法中，首先查找 索引表，然后查找相应的 块表。
35. 一个无序序列可以通过构造一棵 二叉排序树 而变成一个有序序列，构造树的过程即为对无序序列进行排序的过程。
36. 具有 10 个顶点的无向图，边的总数最多为 45。
37. 已知图 G 的邻接表如图所示，其从顶点  $v_1$  出发的深度优先搜索序列为  $v_1v_2v_3v_6v_5v_4$ ，其从顶点  $v_1$  出发的广度优先搜索序列为  $v_1v_2v_5v_4v_3v_6$ 。



38. 索引是为了加快检索速度而引进的一种数据结构。一个索引隶属于某个数据记录集，它由若干索引项组成，索引项的结构为 关键字 和 关键字对应记录的地址。

39. Prim 算法生成一个最小生成树每一步选择都要满足 边的总数不超过  $n - 1$ ，当前选择的边的权值是候选边中最小的，选中的边加入树中不产生回路 三项原则。
40. 在一棵  $m$  阶 B 树中，除根结点外，每个结点最多有  $m$  棵子树，最少有  $m/2$  棵子树。

### 三、判断题。

1. 在决定选取何种存储结构时，一般不考虑各结点的值如何。(✓)
2. 抽象数据类型 (ADT) 包括定义和实现两方面，其中定义是独立于实现的，定义仅给出一个 ADT 的逻辑特性，不必考虑如何在计算机中实现。(✓)
3. 抽象数据类型与计算机内部表示和实现无关。(✓)
4. 顺序存储方式插入和删除时效率太低，因此它不如链式存储方式好。(×)
5. 线性表采用链式存储结构时，结点和结点内部的存储空间可以是不连续的。(×)
6. 对任何数据结构链式存储结构一定优于顺序存储结构。(×)
7. 顺序存储方式只能用于存储线性结构。(×)
8. 集合与线性表的区别在于是否按关键字排序。(×)
9. 线性表中每个元素都有一个直接前驱和一个直接后继。(×)
10. 线性表就是顺序存储的表。(×)
11. 取线性表的第 1 个元素的时间同 1 的大小有关。(×)

12. 循环链表不是线性表。(X)
13. 链表是采用链式存储结构的线性表, 进行插入、删除操作时, 在链表中比在顺序表中效率高。(√)
14. 双向链表可随机访问任一结点。(X)
15. 在单链表中, 给定任一结点的地址 p, 则可用下述语句将新结点 s 插入结点 p 的后面:  $p \rightarrow next = s; s \rightarrow next = p \rightarrow next;$  (X)
16. 队列是一种插入和删除操作分别在表的两端进行的线性表, 是一种先进后出的结构。(X)
17. 串是一种特殊的线性表, 其特殊性体现在可以顺序存储。(X)
18. 长度为 1 的串等价于一个字符型常量。(X)
19. 空串和空白串是相同的。(X)
20. 数组元素的下标值越大, 存取时间越长。(X)
21. 用邻接矩阵法存储一个图时, 在不考虑压缩存储的情况下, 所占用的存储空间大小只与图中结点个数有关, 而与图的边数无关。(√)
22. 一个广义表的表头总是一个广义表。(X)
23. 一个广义表的表尾总是一个广义表。(√)
24. 广义表(((a), b), c) 的表头是((a), b), 表尾是(c)。(√)
25. 二叉树的后序遍历序列中, 任意一个结点均处在其孩子结点的后面。(√)
26. 度为 2 的有序树是二叉树。(X)
27. 二叉树的前序遍历序列中, 任意一个结点均处在其孩子结点的前面。(√)
28. 用一维数组存储二叉树时, 总是以前序遍历顺序存储结点。(X)
29. 若已知一棵二叉树的前序遍历序列和后序遍历序列, 则可以恢复该二叉树。(X)
30. 在哈夫曼树中, 权值最小的结点离根结点最近。(X)
31. 强连通图的各项点间均可达。(√)
32. 对于任意一个图, 从它的某个结点进行一次深度或广度优先遍历可以访问到该图的每个顶点。(X)
33. 在待排序的记录集中, 存在多个具有相同键值的记录, 若经过排序, 这些记录的相对次序仍然保持不变, 称这种排序为稳定排序。(√)
34. 在平衡二叉树中, 任意结点左子树的高度差(绝对值)不超过 1。(√)
35. 拓扑排序是按 AOE 网中每个结点事件的最早发生时间对结点进行排序。(X)
36. 冒泡排序算法关键字比较的次数与记录的初始排列次序无关。(X)
37. 对线性表进行折半查找时, 要求线性表必须以链式方式存储, 且结点按关键字有序排列。(X)
38. 散列表存储的思想是由关键字值决定数据的存储地址。(√)
39. 二叉树为二叉排序树的充分必要条件是任一结点的值均大于其左孩子的值、小于其右孩子的值。(X)
40. 具有 n 个结点的二叉排序树有多种, 其中树高最小的二叉排序树是最佳的。(√)
1. 直接选择排序算法在最好情况下的时间复杂度为  $O(n)$ 。(X)

综合训练题

数据结构

一、单项选择题 (30分)

- 在 C 语言中访问一个顺序存储的线性表, 与  $a[i]$  等价的表示是:
  - A.  $*(a+i)$
  - B.  $\&a[0]+i$
  - C.  $*a+i$
  - D.  $\&a+i$
- n 个结点的完全二叉树中, 叶子结点数是:
  - A.  $n - \lfloor n/2 \rfloor$
  - B.  $\lfloor n/2 \rfloor$
  - C.  $\lfloor n/2 \rfloor + 1$
  - D.  $n - \lfloor n/2 \rfloor + 1$
- 用二叉链表存储二叉树, 则二叉链表中空链域与非空链域的数目之差为:
  - A. 2
  - B. 1
  - C. 0
  - D. 无法确定
- 若要以  $O(n \log_2 n)$  的时间复杂度和  $O(1)$  的空间复杂度进行排序, 则需要用:
  - A. 插入排序
  - B. 堆排序
  - C. 快速排序
  - D. 基数排序
- 为了建立相对平衡的二叉排序树, 输入的结点关键字值最好按什么顺序输入:
  - A. 基本有序
  - B. 随机
  - C. 升序
  - D. 降序
- 图的广度优先遍历借助于下列那种结构来实现的:
  - A. 线性表
  - B. 栈
  - C. 队列
  - D. 二叉树
- 下列用于动态查找表的方法是:
  - A. 顺序查找
  - B. 折半查找
  - C. 二叉查找树
  - D. 分块查找
- 关于 n 个结点连通网的最小生产树描述错误的是:
  - A. 含 n-1 条边
  - B. 没有回路
  - C. 含权值最小的 n-1 条边
  - D. 含 n 个结点
- 若让 a, b, c 依次入栈, 那么下列那种出栈次序不会出现:
  - A. cba
  - B. cab
  - C. bac
  - D. acb
- 下列那种文件组织方法类似于哈希表的实现方式:
  - A. 顺序结构
  - B. 计算地址结构
  - C. 索引结构
  - D. 表结构
- 结点数为 n 的完全二叉树深度至少是:
  - A.  $\lfloor \log_2 n \rfloor - 1$
  - B.  $\lfloor \log_2(n-1) \rfloor$
  - C.  $\lfloor \log_2(n+1) \rfloor$
  - D.  $\lfloor \log_2 n \rfloor + 1$
- 哪种树满足从任意结点出发到根的路径上所经过的结点序列有序:
  - A. 二叉排序树
  - B. 赫夫曼树
  - C. AVL 树
  - D. 堆
- 由权值为 3, 8, 6, 2, 5 的叶子结点生成一颗赫夫曼树, 它的带权路径长度 WPL 为:
  - A. 24
  - B. 48
  - C. 72
  - D. 53
- 在 AOE 网中, 确定某一活动是否位于关键路径上的方法是:
  - A.  $ve=v_l$
  - B.  $ee=e_l$
  - C.  $ve=ee$
  - D.  $v_l=e_l$
- 查找概率相等的 n 个记录构成的查找表的平均查找长度为:
  - A.  $(n-1)/2$
  - B.  $(n+1)/2$
  - C.  $\lfloor n/2 \rfloor$
  - D.  $\lfloor n/2 \rfloor + 1$

2, 12, 10, 16, 30, 8, 28

二、给出一组关键字 (12, 2, 16, 30, 8, 28) 写出用下列算法从小到大排序时一趟结束时的序列: 1) 快速排序 2) 二路归并排序 3) 基数排序 (9分)

三、试求按关键字序列 (12, 1, 4, 3, 7, 8, 10, 2) 插入生成的二叉排序树和平稳二叉树 (在插入过程中进行平衡旋转操作)。画出示意图 (8分)

四、序列 (12, 70, 33, 65, 24, 56, 48, 92, 86, 33) 是否为大顶堆? 若不是请调整为大顶堆, 画出调整前后的树型示意图。画出一趟堆排序后的示意图。(9分)

五、给定二叉树 T, 使用类 C 语言完成下面的算法: (8分)

1) 使用先序遍历求树中所有结点数 C 以及叶子结点数 CO;  
2) 使用后序遍历求该二叉树深度:

六、某一森林由以下双亲-孩子有序对给出: (H, A), (A, B), (B, C), (B, D), (B, E), (H, F), (F, G), (H, H), (H, I), (H, J): (8分)

1) 森林的深度是多少?  
2) 写出中序遍历森林的序列?  
3) 转化为相应的二叉树:  
4) 写出 3) 中二叉树的中序遍历序列:

七、对于下面的 AOE 网, 活动的单位为天。要求: (8分)

1) 整个工程的完成至少需要多久。  
2) 写出关键路径上的活动。

八、假设用于通讯的电文只有 8 种字符 a, b, c, d, e, f, g, h, 其使用频率分别为 0.02, 0.03, 0.06, 0.07, 0.10, 0.19, 0.21, 0.32, 试设计 Huffman 编码。要求画出 Huffman 树, 并写出每个字符的二进制编码。假设在构造 Huffman 树过程中, 权值小的树根形成左子树且编码为 0。(8分)

九、已知一组关键字为 (07, 15, 20, 31, 33, 49, 53, 64, 76, 82) 要构造哈希表, 设表长 m=13, 哈希函数  $H(key) = key \% 13$ : (12分)

1) 求该哈希表的装填系数:

2) 按开放定址线性探测法处理冲突, 构造哈希表, 画出存储示意图;

3) 按链地址法处理冲突, 构造哈希表, 画出存储示意图;

4) 计算 2)、3) 中构造的哈希表的平均查找长度。

十、试写出利用非递归算法 (2 路归并算法) 进行归并排序的算法。(选做题 8 分)

### 参考答案暨评分标准

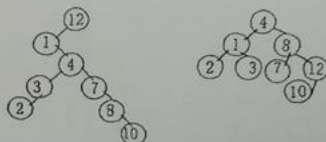
一、单项选择题 (30分=2分×15)

A. A. A. B. B. C. C. C. C. B. B. D. D. D. B. B.

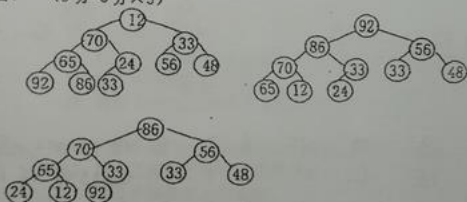
二、 (9分=3分×3)

- 1) {8 2 } 12 {30 16 28}
- 2) 2 12 16 30 8 28
- 3) 30 12 2 16 8 28

三、 (8分=4分×2)



四、 (9分=3分×3)



五、 (8分=4分×2)

```

1) void Count(BiTree T,int &C,int C0){
 if(!T)return;
 C++;
 if(!T->lchild && !T->rchild)C0++
 count(T->lchild,v);
 count(T->rchild,v);
}

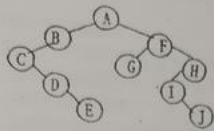
2) int TreeDepth(BiTree T)
{
 if(!T)return 0;
 hL= TreeDepth(T->lchild);
 hR= TreeDepth(T->rchild);
 if hL>hR then return hL+1;
 else return hR+1;
}

```

六、 (8分=2分×4)

1) 3

- 2) CDEBAGFIJH  
3)

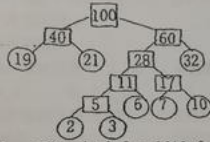


- 4) CDEBAGFIJH

七、对于下面的AOE网,活动的单位为天,要求:(8分)

- 3) 22.  
4) a2 a4 a6 a9 a11

八、(8分=4分×2)



a 10000 b 10001 c 1001 d 1010 e 1011 f 00 g 01 h 11

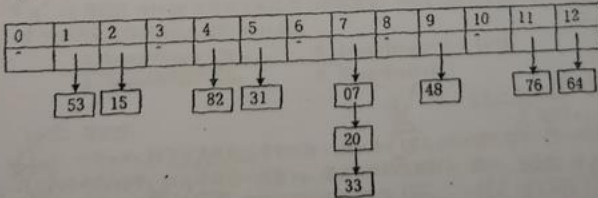
九、(12分=2分+4分+4分+2分)

- 1)  $\alpha=0.77$  (10/13)

2)

|   |    |    |   |    |    |   |    |    |    |    |    |    |
|---|----|----|---|----|----|---|----|----|----|----|----|----|
| 0 | 1  | 2  | 3 | 4  | 5  | 6 | 7  | 8  | 9  | 10 | 11 | 12 |
|   | 53 | 15 |   | 82 | 31 |   | 07 | 20 | 33 | 48 | 76 | 64 |
|   |    |    |   |    |    |   | 20 | 33 | 48 |    |    |    |
|   |    |    |   |    |    |   | 33 |    |    |    |    |    |

- 3)



- 4)

|     |    |    |    |    |    |    |    |    |    |    |     |
|-----|----|----|----|----|----|----|----|----|----|----|-----|
|     | 07 | 15 | 20 | 31 | 33 | 48 | 53 | 64 | 76 | 82 | ASL |
| 线形  | 1  | 1  | 2  | 1  | 3  | 2  | 1  | 1  | 1  | 1  | 1.4 |
| 链地址 | 1  | 1  | 2  | 1  | 3  | 1  | 1  | 1  | 1  | 1  | 1.3 |

十、(8分=2分+3分+3分 选做题)  
void merge(RcdType SR[],RcdType TR[],int l, int m,int n)

```

 {
 k=1;
 for(i=1,j=m+1; i<=m,j<=n;k++)
 {
 if(SR[i]<SR[j]) TR[k]=SR[i++];
 else TR[k]=SR[j++];
 }
 while(i<=m)TR[k++]=SR[i++];
 while(j<=n)TR[k++]=SR[j++];
 }
void mergepass(RodType SR[],RodType TR[],int s,int n)
{
 i=1;
 while(i+2*s-1<=n) {
 merge(SR,TR,i,i+s-1,i+2*s-1);
 i+=2*s;
 }
 if(i+s-1<n) //最后还有两个集合，但不等长
 merge(SR,TR,i,i+s-1,n);
 else //最后还有一个集合
 while(i<=n){TR[i]=SR[i];i++};
}

void mergesort((Sqlist &L)
{
 RodType TR[L.length+1];
 s=1;
 while (s<n) {
 mergepass(L,r,TR,s,n);
 s*=2;
 mergepass(TR,L,r,s,n);
 s*=2;
 }
}

```

一、选择题 (每个空只有一个正确选项)

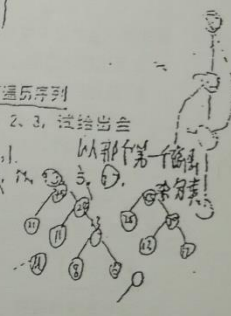
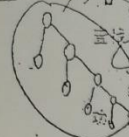
1. 数据结构的三个方面包括 (D).
  - A. 操作对象、逻辑结构和数据映像
  - B. 逻辑结构、存储结构和处理运算
  - C. 顺序结构、链式结构和索引结构
  - D. 算法、算法分析和算法描述
2. 在数据操作中, 从逻辑上可以把数据结构分为 (C).
  - A. 动态结构和静态结构
  - B. 线性结构和非线性结构
  - C. 简单结构和复杂结构
  - D. 内部结构和外部结构
3. 在具有  $n$  个结点的二叉树中, 编号最大的叶子结点的序号是 (A).
  - A.  $[n/2]$
  - B.  $[n/2] \times 2$
  - C.  $[n/2] + 1$
  - D.  $[n/2] + 1 \times 2$
4. 线性表用链式存储结构存储时, 要求内存中可用存储单元 (A).
  - A. 不一定连续的
  - B. 部分地址必须是连续的
  - C. 必须是连续的
  - D. 一定是不连续的
5. 一个栈的入栈序列是 1, 2, 3, 4, 则栈的不可能的输出序列是 (D).
  - A. 1, 2, 4, 3
  - B. 2, 1, 3, 4
  - C. 1, 4, 3, 2
  - D. 4, 3, 2, 1
6. 采用非递归遍历二叉树的深度优先搜索遍历算法类似于二叉树的 (C).
  - A. 按层遍历
  - B. 前序遍历
  - C. 中序遍历
  - D. 后序遍历
7. 若物理一棵具有  $n$  个结点的二叉树平衡, 则其深度不会超过 (B).
  - A.  $n/2$
  - B.  $\log_2 n$
  - C.  $(n+1)/2$
  - D.  $n+1$
8. 带头结点的单链表 head 为空的判定条件是 (D).
  - A. head == NULL
  - B. head->next == NULL
  - C. head->next == head
  - D. head == NULL
9. 在一个单链表中, 若  $p$  结点不是终端结点, 在  $p$  之后插入  $s$  结点, 则执行 (B).
  - A.  $s->next=p; p->next=s;$
  - B.  $s->next=p->next; p->next=s;$
  - C.  $s->next=p->next; p=s;$
  - D.  $p->next=s; s->next=p;$
10. 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, 则执行出队操作时队头指针 front 值为 (A).
  - A.  $front=(front+1) \% m$
  - B.  $front=(front+1) \% (m-1)$
  - C.  $front=(front-1) \% m$
  - D.  $front=front+1$

二、填空题:

1. 一个  $n \times n$  的下三角矩阵 A 中的元素  $a_{ij}$  ( $1 \leq i, j \leq n$ ) 按行等于一个一维数组  $a[1, n(n+1)/2]$  中, 对其中的任一元素  $a_{ij}$ , 它在 B 中的位置为  $k$ , 则  $k = \frac{i(i-1)}{2} + j$ .
2. 两个长度分别为  $m$  和  $n$  ( $m < n$ ) 的升序的表归并成一个升序的表, 最少要进行  $\min(m, n)$  次比较.
3. 设二维数组:  $array[-1..4, 0..3]$  of integer, 每个元素 (数组) 占 2 个存储单元, 元素按行的顺序存储, 数组的起始地址为 200, 元素  $w[2, 1]$  的地址是  $200 + (2+1) \times 2 = 226$ .
4. 线性表  $L = (a_1, a_2, \dots, a_n)$  采用顺序存储, 任意不同的  $nt$  个位置上插入的概率



1. 已知一个字符串为 "11, 3, 9, 12, 32, 4, 66, 7, 32, 65, 100", 当二叉  
 查找树为 32 的查找树, 依次比较后查找成功, 第 4 个  
 2. 在一个无向图的邻接表中, 若某结点的个数是  $n$ , 则图中边的条数是  $n$  条。  
 3. 设一个函数列表的变量为  $M$ , 用线性探测法解决冲突, 要插入一个键值, 需插入成  
 功, 最多要进行  $M$  次比较。



三、应用题:

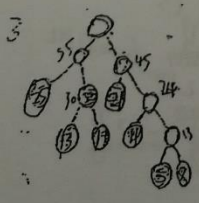
- 已知一棵二叉树如图(a)所示, 分别写出该树的前、中、后序遍历序列
- 对于一个数, 给出输入项 1, 2, 3, 若是输入(入栈)顺序为: 1, 2, 3, 试给出会  
 部可能的输出(出栈)序列。 1, 2, 3; 2, 1, 3; 1, 3, 2; 2, 3, 1; 3, 2, 1
- 已知一正文由 7 种字符组成, 字符在正文中出现的频率分别是 7, 14, 21, 28, 35, 42, 49, 用这 7 个字符设计哈夫曼树和哈夫曼编码 (6 分)。
- 续有如图(b)的一元问题, 完成下述要求: (5 分)
  - 若按顺序输入顶点为: (1, 3), (2, 3), (3, 4), (1, 2), (3, 5)  
 (4, 1), (2, 5) 用头插法画出邻接表。
  - 分别写出从顶点 v4 为出发点的 DFS 和 BFS 序列。 4 1 2 3 5
  - 按普里姆算法画出从图从 v1 出发的一棵最小生成树。 4 1 3 2 5
- 已知关键字序列 {40, 17, 60, 18, 73, 7, 85, 24, 65, 22}, 写出快速排序的每  
 一趟排序结果(按升序); 写出建成小堆的每一步结果。
- 设有一组关键字 (给新函数)  $H(\text{key}) = (3 * \text{key}) \% 11$ , 用线性探测法处理冲突, 试  
 在 0-10 的散列(哈希)地址空间中对该关键字序列 {14, 23, 01, 33, 20, 35, 27,  
 55, 32, 12}, 构造散列表, 并求等概率情况下查找成功和不成功的平均查找长度。
- 已知给定的二叉树是森林变换过来的。
  - 将该二叉树还原成森林;
  - 写出森林的先根遍历序列和后根遍历序列。

1 -> 4 -> 3  
 2 -> 5 -> 1  
 3 -> 5 -> 4 -> 2  
 4 -> 1 -> 3  
 5 -> 2 -> 3

7 14 21 28 35 42 49  
 40 17 60 18 73 7 85 24 65 22  
 14 23 01 33 20 35 27 55 32 12

算法设计  
 已知一个以 head 为头结点指针的单链表, 请用冒泡排序法对链表按数据域值从小  
 到大进行排序。  
 已知存在一棵以 T 为根结点指针的二叉树, 试写一个递归算法 Search (T, k)  
 查找树中结点关键字为 k 的结点, 若找到则返回其结点地址, 否则返回空指针。并  
 分析其时间复杂度。

2. ① 1, 2, 3 ② 2, 1, 3 ③ 3, 2, 1 ④ 2, 3, 1 ⑤ 1, 3, 2



4: 8 111  
 5: 11 110  
 6: 13 010  
 7: 110 1  
 8: 17 011  
 9: 25 100

```

Search (T, k) {
 if (!T || p! = NULL) return P;
 if (T->data == k) p = bt;
 else {
 Search (k < T->lchild, k);
 Search (k > T->rchild, k);
 }
 if (!T) return NULL;
 if (T->data == k) return T;
 else if (!search (T->lchild, k))
 return search (T->rchild, k);
 else return search (T->lchild, k);
}

```

4. 设计算法以判断有  $n$  个顶点的无向图  $G$  是否是 2-图。若是，则返回 true，否则返回 false。

(注：本算法中可以调用以下几个函数：

- $\text{firstadj}(G, v)$ ——返回图  $G$  中顶点  $v$  的第一个邻接点的序号，若不存在，则返回 0；
- $\text{nextadj}(G, v, f)$ ——返回图  $G$  中顶点  $v$  的邻接点中比  $f$  之后邻接点的序号，若不存在，则返回 0；

另外，如果涉及邻接点列表操作，可直接采用邻接列表的形式，不必写出其实现形式)

5. 已知程序没有数组  $\text{data}$  和整型变量  $\text{listlen}$  两个分量，前者存储范围从 0 到  $n$ ，用于存放元素值，后者记录表中的实际元素个数，并已知  $\text{data}$  中的元素按从小到大的次序排列，设计算法以删除表中重复的元素，并要求时间复杂度为  $O(n)$ ，例如，对表  $\{1, 1, 2, 3, 3, 4, 6, 8, 9\}$  执行的结果为  $\{1, 2, 3, 4, 6, 8, 9\}$ 。

中国科技大学数据结构和算法期末考试试题 (2003.1)

学号: \_\_\_\_\_ 姓名: \_\_\_\_\_ 成绩: \_\_\_\_\_

一、填空题(24分, 每空 1.5分)

1. 假设一循环队列用数组  $A[0..m-1]$  存放元素值, 已知其头尾指针分别是  $\text{front}$  和  $\text{rear}$  (指向队尾元素的下一个), 则当前队列中元素的个数是  $(\text{rear} - \text{front} + m) \% m$

2. 队和队列均是线性表的线性表, 队的特点是 先进后出, 队列的特点是 先进先出

若已知一个栈的入栈序列是  $1, 2, 3, \dots, n$ , 其输出序列为  $p_1, p_2, p_3, \dots, p_n$ , 若  $p_1 = n$ , 则  $p_i (1 \leq i \leq n-1)$  为  $n-i+1$

3. 具有  $n$  个结点的  $k$  ( $k \geq 2$ ) 的二叉树中, 至少有  $2k-1$  个结点, 至多有  $2^k - 1$  个结点,  $\frac{\log_2(n+1)}{\log_2 k}$

4. 具有 5 个结点的不同形态的二叉树有 8 棵, 具有 5 个结点的不同形态的树有 34 棵。

5. 在平衡二叉树中插入一个结点后造成了不平衡, 设最低的不平衡结点为  $A$ , 若  $A$  的左孩子的平衡因子为 -1, 右孩子的平衡因子为 0, 则应旋 左 调整使其恢复平衡。

6. 高度为 5 的完全二叉树中至少有 15 个叶子, 至多有 15 个结点有右孩子。

7. 具有 8 个顶点的无向图至少应有 7 条边才能确保它是一个连通图, 至多有 21 条边才能确保它仍是一个非连通图。

8. 判断一个有向图是否存在回路除了可以用拓扑排序方法外, 还可以用 DFS 方法。

9. 广义表  $((x), ()), ((a, b), a), b$  的表头是  $((x), ())$ , 表尾是  $((a, b), a), b$

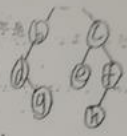
二、解答题(41分)

1. 试推导出  $n$  个结点的完全  $k$  叉树的深度  $H$ 。(5分)

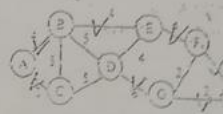
$\lceil \log_k(n+1) \rceil$  ;  $\lceil \log_k(nk-n) \rceil + 1$

2. 某二叉树的前序遍历结点访问顺序是 a b d g c e f h, 中序遍历的结点访问顺序是 d g b e a c h f, 试画出其后序遍历的结点访问顺序, 并画出该二叉树。(8分)

g a b e h f c a



3. 画出右图所示向图中的最小生成树和深度优先生成树(从 A 出发)。(10分)



4. 画出先序遍历序列为 [45, 24, 53, 12, 30, 5, 16, 18]

的二叉树序列以及在此基础上删除关键字 24 之后的二叉树序列。(10分)

5. 已知字母 A, B, C, D, E, F, G 出现的概率分别是 0.03, 0.02, 0.16, 0.12, 0.15, 0.28, 0.30, 试为其设计哈夫曼编码。(8分)

A: 11111  
B: 11110  
C: 1110  
D: 110  
E: 10  
F: 00  
G: 01

### 三、算法设计(35分)

1. 现有一大块空闲区(简称被管理内存区)可用于动态存储管理, 请按如下管理模式为其编写初始化 xinit, 分配 xalloc 和释放 xfree 函数。(20分)

1) 系统为被管理内存区引入一个目录信息, 其信息结构定义如下:

```
typedef struct area {
 char *startptr; // 被管理内存区的起始地址
 int size; // 被管理内存区的大小
 char *curptr; // 被管理内存区当前可分配的起始地址
} Area;
Area areainfo; // 当前的被管理内存区的目录信息
```

2) 在实际运行中, 被管理内存区或者是一整块空闲区; 或者被分成两部分: 前半部为已分配区(由多次分配的区域顺序组成), 后半部为空闲区(起始位置存放在 areainfo.curptr 中)。

3) 引入一个双向循环链表(已分配目录表, 其结点空间通过 malloc 获得, 不占用被管理内存区)来记载已分配区中的各次分配情况, 每一次分配对应于表中的一个结点, 该结点记载相应分配区的目录信息, 存储结构如下:

```
typedef struct node {
 struct node *prev, *next; // 直接前驱结点和直接后继结点的指针
 char *startptr; // 该次分配的内存区起始地址
 int size; // 该次分配的内存区的大小(字节数)
 char flag; // 0-在用, 1-不再使用, 但空间尚未回收
```

} Node; \*DLink;

每当分配请求时, 根据所需分配的 size, 从 areainfo.curptr 指向的空闲区中申请, 若当前空闲区内存区剩余的区段已不相等, 则返回 NULL; 否则修改 areainfo.curptr, 且已分配目录表的表尾增加一个记载当前被分配区的目录信息的结点(flag 置为 0), 并返回当前被分配区的起始地址。

当释放某内存区时, 若该内存区不是最近分配的, 则将该内存区对应的目录信息置为 flag 置为 1 (不是最近分配的), 否则回收该区域, 释放对应的目录信息; 若该内存区被释放的分配区也已不再使用, 则一并回收。

2. 试写一个算法, 以邻接矩阵方式存储的有向图 G 中来求点  $v_i$  到点  $v_j$  的长度为 k 的最短路径。(15分)