

试卷 7

一、填空题(每空 2 分, 共 30 分)

1. _____是数据的基本单位, 在计算机程序中通常作为一个整体进行考虑和处理。
2. 数据的逻辑结构可分为集合、线性结构、_____结构和_____结构。
3. 数据元素之间的关系在计算机中有两种不同的表示方法: 顺序映象和非顺序映象, 由此得到两种不同的存储结构: 顺序存储结构和_____。
4. 算法的五个重要特性包括: 有穷性、____、____、输入和输出。
5. 下面的算法计算实数 x ($x > 0$) 的非负整数 n ($n \geq 0$) 次幂, 其时间复杂度是_____。

```
double Power(double x, int n)
{
    double y = 1;
    if (n > 0)
    {
        y = Power(x, n / 2);
        y *= y;
        if (n % 2 == 1) y *= x;
    }
    return y;
}
```

6. 只在表的一端进行插入和删除的线性表称为_____。在表的一端进行插入、另一端进行删除的线性表称为_____。
7. 在 C 语言中定义下面的二维实型数组:
double a[5][10];
每个元素占用 8 字节内存空间, 若数组起始地址为 0x1000, 则元素 a[3][5] 的地址为 0x_____。
8. 已知完全二叉树有 1024 个结点, 则该二叉树的深度为_____。

9. 包含 5 个顶点的有向图, 至多有_____条弧。
10. 深度为 10 的完全二叉树至少有_____个结点, 至多有_____个结点。
11. 包含 20 个顶点的连通图, 其最小生成树拥有_____条边。

二、单项选择题 (每小题 2 分, 共 20 分)

1. 若元素 1, 2, 3 依次进栈, 则出栈的次序不可能是_____。
A) 3, 2, 1 B) 1, 3, 2 C) 3, 1, 2 D) 2, 1, 3
2. 在无附加头结点的链栈中, 若栈顶指针为 top, 将指针 s 所指示的结点入栈, 所执行的操作为_____。
A) s->next = top; top = top->next;
B) top = s; s->next = top;
C) s->next = top->next; top->next = s;
D) s->next = top; top = s;
3. 在无附加头结点的链栈中, 若栈顶指针为 top, 则判断栈空的条件是_____。
A) top==NULL B) top!=NULL
C) top->next == top D) top->next==NULL
4. 循环队列存于数组 a[M]中, 假定队首和队尾指针分别是 front 和 rear, 则判断队空的条件是_____。
A) front == 0 B) rear == 0 C) front == M - 1 D) front == rear
5. _____是关于排序算法的错误说法。
A) 直接插入法和冒泡法是稳定的
B) 快速法在任何情况下都是最快的
C) 二路归并算法需要的辅助空间是 $O(n)$
D) 堆排序的时间复杂度是 $O(n \lg n)$
6. 主关键字是指能唯一标识一条记录的_____。
A) 一个数据项 B) 一组数据项 C) A 或 B D) 以上都不是
7. 按照二叉树的定义, 具有 3 个结点的二叉树有_____种形态。

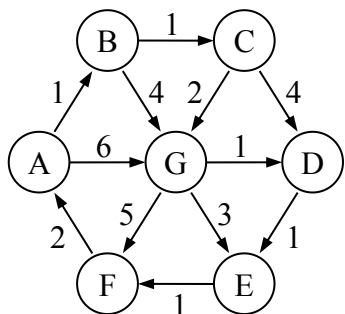
- A) 3 B) 4 C) 5 D) 6
8. 求串 s2 在串 s1 中首次出现的位置的运算是_____。
- A) 连接 B)求串长 C) 求子串 D) 模式匹配
9. 以下数据结构中, _____是线性结构
- A) 串 B) 广义表 C) 稀疏矩阵 D)二叉树
10. 下面算法的时间复杂度, _____效率最高
- A) $O(n)$ B) $O(\lg n)$ C) $O(n \lg n)$ D) $O(n^2)$

三、分析题（每小题 5 分，共 30 分）

1. 若二叉树先根遍历的序列为: EFGADCB, 中根遍历的序列为 GDAFBCE, 请画出二叉树形态。
2. 已知电文为“ABBCDADDDCACAAD”, 根据字母的出现频率构造哈夫曼树, 并写出每个字符的哈夫曼编码。

字母	频率	哈夫曼编码
A		
B		
C		
D		

3. 请写出用 Dijkstra 算法求从顶点 1 出发到其余顶点的最短路径的计算过程。



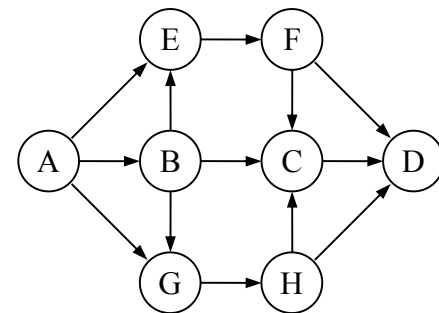
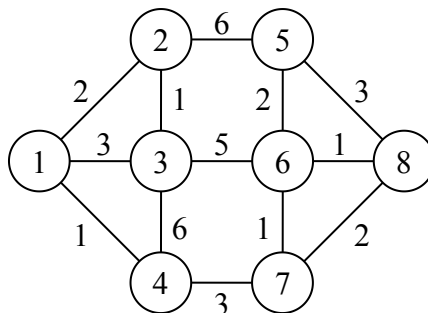
4. 已知散列表表长为 15, 地址计算公式为

$$H(k) = k \bmod 13$$

冲突处理方式为线性探测再散列, 将关键字 5、15、18、2、3、31、16、4 依次插入到散列表中, 请写出散列表的状态。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

5. 根据下面左边图所示的无向图, 请用 Prim 算法从顶点 1 出发求最小生成树的步骤。



6. 根据上面右图所示的有向无环图进行拓扑排序, 请写出至少 5 种排序结果。

四、综合应用题（每小题 10 分，共 20 分）

1. 单值化处理

顺序表结构类型定义如下:

```
struct ALIST
{
    float *data;    // 动态数组起始地址
    int size, length; // 动态数组大小、顺序表长度
};
```

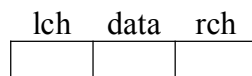
请编写算法, 对于顺序表中所有值相同的元素, 只保留其中第一个元素, 删除其余的元素。

假如线性表为(2.5, 3.8, 1.2, 2.5, 4.7, 3.8, 3.8, 9.1, 1.2), 则经过单值化处理后将变为(2.5, 3.8, 1.2, 4.7, 9.1)。

```
void Unique(ALIST *list);
```

要求：用文字描述算法思想，并估算时间复杂度，然后用 C/C++ 语言编码。

2. 二叉树采用链式存储结构，结点的存储结构如下图所示：



结点的结构类型定义如下：

```
struct NODE
```

```
{  
    float data;           // 数据域  
    struct NODE *lch, *rch; // 指针域  
};
```

请编写算法求二叉树的深度。

```
int Depth(NODE *root);
```

函数的参数是根指针，函数值是二叉树的深度。如：

```
printf("%f\n", Depth(root));
```

要求：用文字描述算法思想，并估算时间复杂度，然后用 C/C++ 语言编码。

数据结构试卷 7 答案

一、填空题(每空 2 分，共 30 分)

- 3. 数据元素
- 4. 树型、图状(或网状)
- 5. 链式存储结构
- 6. 确定性、可行性 (或可行性、确定性)
- 7. $O(\lg n)$
- 8. 栈、队列
- 9. 0x1118
- 10. 11
- 11. 20
- 12. 512、1023
- 13. 19

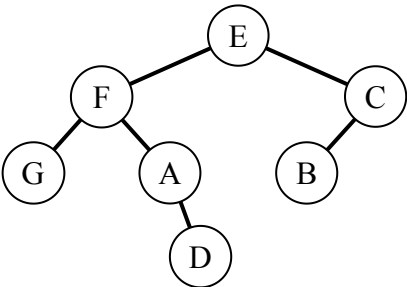
二、单项选择题 (每小题 2 分，共 20 分)

- 14. C
- 15. D
- 16. A
- 17. D
- 18. B
- 19. C
- 20. C

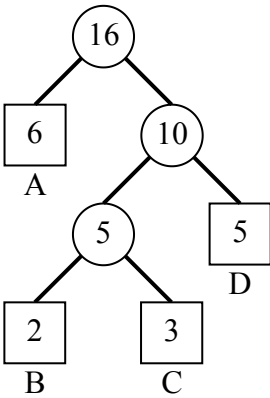
- 21. D
- 22. A
- 23. B

三、分析题（每小题 5 分，共 30 分）

24.



25.



字母	频率	哈夫曼编码
A	6	0
B	2	100
C	3	101
D	5	11

26.

步骤	顶点	距离					路径				
		A	B	C	D	E	A	B	C	D	E
0	A	0	1	∞	6	4	-	A	-	A	A
1	B	0	1	9	6	3	-	A	B	A	B
2	E	0	1	8	4	3	-	A	E	E	B

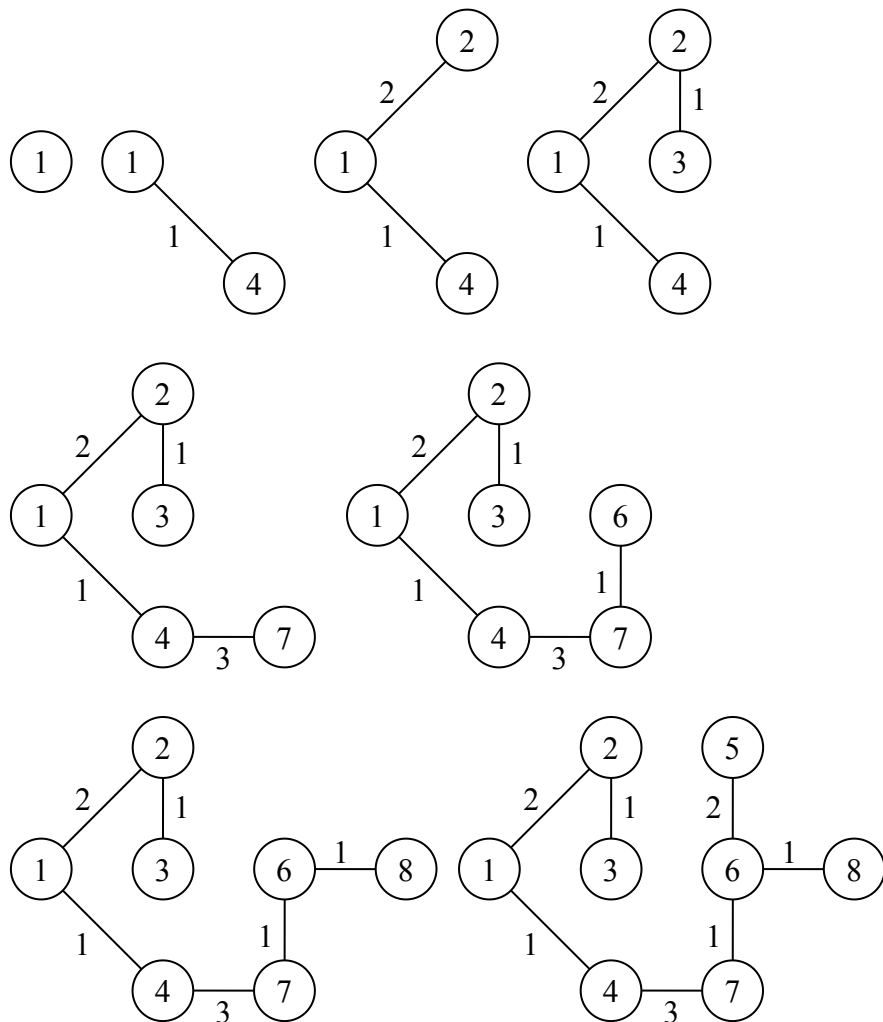
路径	长度	最短路径
A→B	1	A→B
A→C	7	A→B→E→D→C
A→D	4	A→B→E→D
A→E	3	A→B→E

3	D	0	1	7	4	3	-	A	D	E	B
4	C	0	1	7	4	3	-	A	D	E	B

27.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		15	2	3	5	18	31	16	4					

28.



29. ABIEFGHCD
ABEGHFCD

ABIEFGHCD
ABGHEFCD
ABIEFGHCD

四、综合应用题（每题 10 分，共 20 分）

30. 变量 n 表示要删除的元素的数量。遇相同的值，则 n 加 1；遇不同的值，则将该元素前移 n 个位置，最后将表长减去 n 。

```
void Unique(ALIST *list)
{
    int i, j, n;
    for (i = 0; i < list->length; i++)
    {
        for (n = 0, j = i + 1; j < list->length; j++)
        {
            if (list->data[j] == list->data[i])
            {
                n++;
            }
            else if (n > 0)
            {
                list->data[j - n] = list->data[j];
            }
        }
        list->length -= n;
    }
}
```

时间复杂度为 $O(n^2)$ 。

31. 空二叉树深度为 0。非空二叉树，先求左、右子树的深度，取最大值再加 1(即根结点)。

```
int Depth(NODE *root)
{
    int d = 0, d1, d2;
    if (root)
    {
        d1 = Depth(root->lch);
        d2 = Depth(root->rch);
        d = (d1 >= d2 ? d1 : d2) + 1;
    }
}
```

```
    return d;  
}
```

时间复杂度为 $O(n)$ 。

求深度的语句若写成

```
d = (Depth(root->lch) >= Depth(root->rch) ? Depth(root->lch) :  
Depth(root->rch)) + 1;
```

则会导致重复计算, 时间复杂度可达 $O(2^n)$, 应扣分。