

# 第一章 计算机基础

## 1.1 数据结构

1. 评估下面这段代码的时间复杂度 ()

```
int func(int n) {  
    int i = 0, sum = 0;  
    while(sum < n) sum += ++i;  
    return i;  
}
```

1  
2  
3  
4  
5

*Solution.*



2. 评估下面这段代码的时间复杂度 ()

```
int sum = 0;  
for(int i = 1; i < n; i *= 2)  
    for (int j = 0; j < i; j++)  
        sum++;
```

1  
2  
3  
4

*Solution.*



3. 一个栈的入栈序列为 $1, 2, 3, \dots, n$ , 出栈序列是 $P_1, P_2, P_3, \dots, P_n$ . 若  $P_2 = 3$ , 则  $P_3$  的可能取值的个数可能是 ()  
A.n-1    B.n-2    C.n-3    D. 无法确认
4. 已知循环队列 存储在一维数组  $A[0, \dots, n-1]$  中, 且队列非空的时候 front 和 rear 分别指向队头和队尾. 若初始时队列为空, 且要求第一个进入队列的元素存储在  $A[0]$ , 则初始

时 front 和 rear 的值分别为 ()

- A.0,0                  B.0,n-1                  C.n-1,0                  D.n-1,n-1

5. 循环队列 放在一维数组  $A[0, \dots, M-1]$  中, end1 指向队头元素, end2 指向队尾元素的后一个位置. 假设队列两端都可以进行入队和出队操作, 队列中最多能容纳  $M-1$  个元素. 初始队列不为空. 下列判断对空和队满的条件中, 正确的是 ()

- A. 对空: end1 == end2;                  队满: end1 == (end2+1) mod M  
B. 对空: end1 == end2;                  队满: end2 == (end1+1) mod M-1  
C. 对空: end2 == (end1+1) mod M;                  队满: end1 == (end2+1) mod M  
D. 对空: end1 == (end2+1) mod M;                  队满: end2 == (end1+1) mod (M-1)

6. 火车重排问题

假设火车入口和出口之间有  $n$  条轨道, 列车驶入的顺序为 8, 4, 2, 5, 3, 9, 1, 6, 7 若希望得到的驶出顺序为 1~9 则  $n$  至少为 ()

- A.2                  B.3                  C.4                  D.5

7. 在一颗度为 4 的树  $T$  中, 若有 20 个度为 4 的结点, 10 个度为 3 的结点, 1 个度为 2 的结点, 10 个度为 1 的结点, 则树  $T$  的叶结点个数为 ()

- A.41                  B.82                  C.113                  D.122

8. 已知一颗完全二叉树的第六层 (设根为第一层) 由 8 个叶结点, 则该完全二叉树的结点个数最多为()

- A.39                  B.52                  C.111                  D.119

9. 若一颗完全二叉树有 786 个结点, 则该二叉树中叶结点的个数为 ()

- A.257                  B.258                  C.384                  D.385

10. 先序序列为  $a, b, c, d$  的不同二叉树的个数为 ()

- A.13                  B.14                  C.15                  D.16

11. 将森林转换为对应的二叉树, 若在二叉树中, 结点  $u$  是结点  $v$  的父结点的父结点, 则在原来的森林中,  $u$  和  $v$  可能的关系是 ()

(I) 父子关系

(II) 兄弟关系

(III)  $u$  的父结点与  $v$  的父结点是兄弟关系

12. 已知一颗有 2011 个结点的树, 其叶结点的个数为 116, 该树对应的二叉树中无右孩子的结点个数为 ( )
13. 已知森林 F 及与之对应的二叉树 T, 若 F 的先根遍历序列为 $a, b, c, d, e, f$ , 中根遍历序列为 $b, a, d, f, e, c$ , 则 T 的后根遍历序列为 ( )
14. 对任意给定的含  $n(n>2)$  个字符的有限集合 S, 用二叉树表示 S 的哈夫曼编码集与定长编码集, 分别得到二叉树  $T_1, T_2$ . 下列叙述中, 正确的是 ( )
- A.  $T_1$  和  $T_2$  的结点个数相同
- B.  $T_1$  的高度大于  $T_2$  的高度
- C. 出现频次不同的字符在  $T_1$  中处于不同的层
- D. 出现频次不同的字符在  $T_2$  中处于相同的层
15. 在由 6 个字符构成的字符集 S 中, 各字符出现的频次为 $3, 4, 5, 6, 8, 10$ , 为 S 构造的哈夫曼树的加权平均长度为 ( )
- A.2.4                  B.2.5                  C.2.67                  D.2.75
16. 对于任意一棵高度为 5 且有 10 个结点的二叉树, 若采用顺序存储结构保存, 每个结点占一个存储单元, 则存放该二叉树至少需要多少存储单元?

**Solution.** 对应顺序存储, 应该按照满二叉树存储, 故需要的存储空间为  $2^h - 1 = 2^5 - 1 = 31$  个 □

17. 在下列关于二叉树遍历的说法中, 正确的是 ( ).
- (A) 若有一个结点是二叉树中某个子树的中序遍历结果序列的最后一个结点, 则它一定是该子树的前序遍历结果序列的最后一个结点
- (B) 若有一个结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点
- (C) 若有一个叶结点是二叉树中某个子树的中序遍历结果序列的最后一个结点, 则它一定是该子树的前序遍历结果序列的最后一个结点
- (D) 若有一个叶结点是二叉树中某个子树的前序遍历结果序列的最后一个结点, 则它一定是该子树的中序遍历结果序列的最后一个结点

**Solution.** 二叉树中序遍历的最后一个结点必然是从根开始沿着起右指针走到底的结点, 记其为  $p$ .

若其不是叶子结点,不妨假设其左子树非空,则在先序遍历中最后的结点必然在其左子树中,故 ABD 都不对.

若其是叶子结点,则前序遍历和中序遍历的最后一个结点都会是它,故 C 正确.  $\square$

## 1.2 计算机网络

1. TCP/IP 参考模型分为四个层次 (正确)

**Solution.** 虽然一直讲物理层-数据链路层-网络层-传输层-应用层,TCP/IP 五层结构;但如果问层次,其实物理层和数据链路层做的事情可以被概况的,通常称为网络接口层.  $\square$

2. 二进制信号在信噪比为 127:1 的 4kHz 的信道上传输,最大数据传输速率可达到 ()

A.28000bps      B.8000bps      C.4000bps      D. 无限大

**Solution.** 考虑奈氏定理,有  $B = 2W = 8kBand$  此时最大数据传输速率为

$$R_{N-max} = B \times \log_2(n) = 8kpbs$$

考虑香农定理,有

$$R_{S-max} = W \times \log_2(1 + S/N) = 4kHz \times \log_2(1 + 127) = 28kpbs$$

最终的传输速率有二者的**最小值**限制,故  $R_{max} = \min(R_{Nmax}, R_{Smax}) = 8kpbs$   $\square$

## 1.3 计算机组成原理

1. 某计算机字长为 8 位,CPU 中有一个 8 位加法器. 已知无符号数  $x=69,y=38$ , 如果在该加法器中计算  $x-y$ , 则加法器的两个输入端入端信息和低位进位信息分别是 ()

A.0100 0101,0010 0110, 0      B.0100 0101,1101 1001, 1

C.A.0100 0101,1101 10110, 0      D.0100 0101,1101 1010, 1

2. 某计算机存储器按字节编制, 采用小端方式存放数据. 假定编译器规定 int 型和 short 型长度分别为 32 位和 16 位并且数据按边界对齐存储. 某 C 语言程序段如下

<code>struct {</code>	1
<code>    int a;</code>	2
<code>    char b;</code>	3
<code>    short c;</code>	4
<code>}record;</code>	5
<code>record.a = 273;</code>	6

若 record 变量的首地址为 0xC008 地址 0xC008 中的内容及 record.c 的地址分别是 ()

A.0x00, 0xC00D      B.0x00,0xC00E      C.0x11,0xC00D      D.0x11,0xC00E

3. 有如下 C 语言序段:

<code>short si = -32767;</code>	1
<code>unsigned short usi = si;</code>	2

这执行上述两条语句后,usi 的值是 \_\_\_\_\_

## 1.4 操作系统