

## 数据结构期中考试题答案解析

1

参考答案：B

本题要求具备一定的代码阅读能力。 $x$ 的初值为91，每执行10次 $x++$ ，就执行一次 $y=2*y$ ，且 $x$ 恢复为初值91。也就是说，每11次循环，会导致 $y=2*y$ 。当 $y \geq n$ 时停止while循环，因此时间复杂度为 $O(11 \log_2 n) = O(\log_2 n)$ 。

2.

参考答案：C

括号匹配的过程中，扫描到左括号，则入栈，扫描到右括号，则弹出栈顶左括号，并与右括号进行匹配。栈S的变化如下：

①{ 入栈，②{ 入栈，③{ 出栈，④( 入栈，⑤[ 入栈，⑥( 入栈，⑦( 出栈，⑧[ 出栈，⑨( 入栈，⑩( 出栈，⑪[ 入栈，⑫[ 出栈，⑬( 出栈，⑭{ 出栈

当运行到第⑥步时，栈内元素最多，共4个，因此栈的大小至少为4

3.

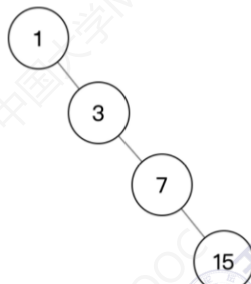
参考答案：D

三对角矩阵按列优先方式压缩存储。 $A[20]$  是第 21 个元素。三对角矩阵的前7列共  $2+3 \times 6=20$  个元素。因此第 21 个元素刚好是第八列的第一个元素，即  $m_{7,8}$

4.

参考答案：D

二叉树的顺序存储，需要将结点与“完全二叉树”对应。当每个分支结点都只有右孩子时，所需要的存储单元数量最多。二叉树的形态如下：



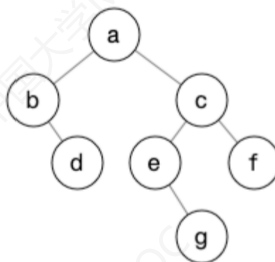
此题可以利用完全二叉树结点编号之间的关系来推导，对于一棵完全二叉树，若根节点编号从1开始，则每个分支节点  $i$  的右孩子编号为  $2i+1$ 。由此可得，最下层的叶子结点对应完全二叉树的结点编号为15。因此，该二叉树需要的存储单元数量最多是15。

当然，也可以用另一个思路来计算， $h$ 层的满二叉树有总共有  $2^h-1$  个结点，因此四层满二叉树总共有 15 个结点。

5.

参考答案：C

由于二叉树的前序遍历和中序遍历序列能确定唯一的一棵二叉树，因此本题可以确定二叉树 T 的结构。

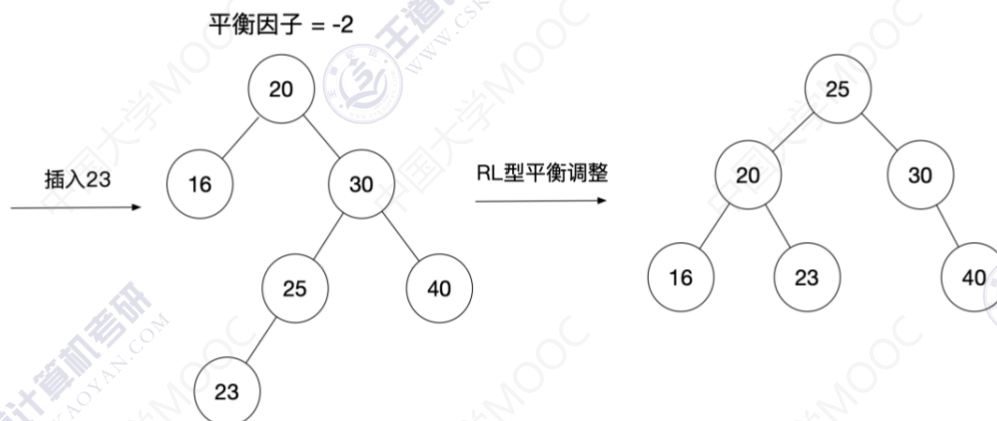


逐层断掉T中最右侧的右子树，即得到三棵树，因此森林F中树的棵树是3。

6.

参考答案：D

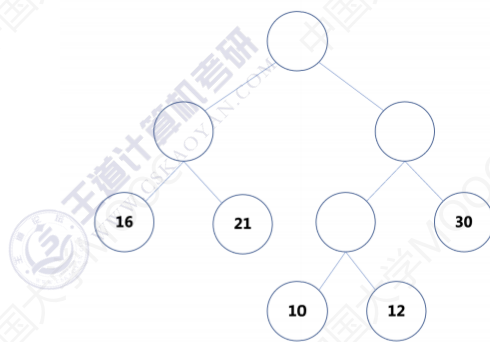
将关键字23插入作为关键字25所在结点的左孩子后，该二叉树失衡，经过平衡调整后，该平衡二叉树变为如下图所示结构，根节点中的关键字为25。



7.

参考答案：B

要求最小的 WPL，即考查哈夫曼树的构造，构造结果如下图所示，因此  $WPL = 2 \times (16 + 21 + 30) + 3 \times (10 + 12) = 200$ 。



8.

参考答案：A

A 选项：该选项不严谨，若各个字符结点的权值都为 1，则所有的结点都是“权值最小的结点”，但不一定都是兄弟。

B 选项：根据哈夫曼树的构造过程，包含  $n$  个叶结点的哈夫曼树，最大高度为  $n$ 。

C 选项：二叉哈夫曼树中，只有度为 2 和度为 0 的结点。度为 0 的结点数为  $n$ ，假设度为 2 的结点

数为  $m$ ，则  $2m+1 = m+n$ ，即  $m=n-1$ 。因此该哈夫曼树的结点总数为  $2n-1$ 。

D 选项：根据哈夫曼树的构造过程可知，单独一个根节点的权值，就等于所有叶子结点的权值之和，再加上其他非叶结点的权值，一定大于所有叶子结点的权值之和。

9.

参考答案：C

处理这个题目，并不要求 next 数组，发生失配的时候，模式串往后滑就行。可以快速做题

S: ABCABCDABABCDABCDABDE

T: ABCDABD

第一次发生失配，累计比较了 4 次字符

S: ABCABCDABABCDABCDABDE

T: ABCDABD

第二次发生失配，累计比较了 4+7 次字符

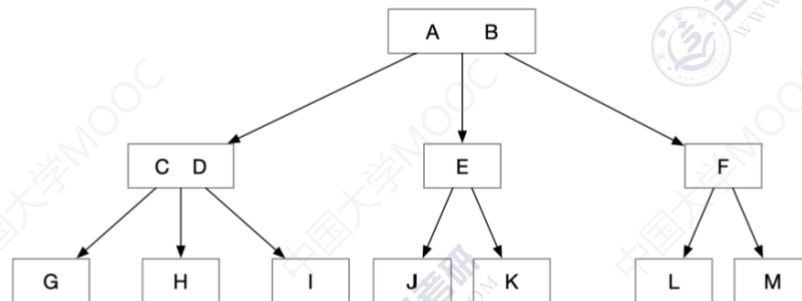
S: ABCABCDABABCDABCDABDE

T: ABCDABD 第三次发生失配，累计比较了 4+7+1 次字符，即累计对比12 次

10.

参考答案：A

阶数为3的B树中，每个结点包含的关键字可以为1个或2个。本题中，要求第二层有4个关键字，则结点数最多的情况如下图所示，其中A、B、C、D... 表示关键字，最多可能有11个结点。



11.

参考答案：D

题目应是一种输出受限的双端队列，因此根据选项 C 和 D，若 4 是第一个出队的元素，那么至少 1, 2, 3, 4 已全部入队，则队列中 2 应该与 1 相邻，出队时顺序也应该相邻，所以选项 D 错误。

### 应用题

1.

参考答案：(1)

可以使用一个数组  $a[26]$  保存 A~Z 是否出现过，0 表示已经未出现过， $>0$  表示出现过。扫描字符串 S，更新数组 a。然后用 num 统计有多少个不同的字母出现。

(2)

```
int ans(char S[],int n) {  
    int j,a[26]={};  
    int num=0; //num统计有几个不同的字母出现  
    for (int i=0;i<n;i++) //扫描串s  
    {  
        j=S[i]-'A'; //j表示是A开始的第几个字母，A是第0个  
        a[j]++;  
    }  
    for (int i=0;i<26;i++) //统计  
        if (a[i]!=0) num++;  
  
    return num;  
}
```

(3) 空间复杂度主要是数组 a[26]，是常数所以是  $O(1)$ 。时间复杂度主要是两个循环，一个循环 n 次，一个循环 26 次，所以是  $O(n)$

2.

参考答案：

(1)

```
typedef struct node  
{  
    int data;  
    struct node *lchild , *rchild;  
} BTree;
```

(2) 当 $h > 2$ 时,  $a[h] = a[h-1] + a[h-2] + 1$

(3) 注: 每个步骤正确得3分

