

## 一、单选题(2 × 10)

1. 用不带头结点的单链表方式存储的队列，在进行插入操作时，( )
  - 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
2. 往一个大小为  $n$  的顺序表中任一位置插入一个新的数据节点时，平均需要移动( )个节点
  - A.  $n$
  - B.  $\frac{n}{2}$
  - C.  $n + 1$
  - D.  $\frac{n+1}{2}$
3. 一个栈的元素的进栈顺序是  $a\ b\ c\ d\ e$ ，那么下面哪一个不能作为出栈顺序？( )
  - A.  $e\ d\ c\ b\ a$
  - B.  $d\ e\ c\ b\ a$
  - C.  $d\ c\ e\ a\ b$
  - D.  $a\ b\ c\ d\ e$
4. 设有一个二维数组  $A[m][n]$ ，假设  $A[0][0]$  存放位置在  $100_{(10)}$ ， $A[3][3]$  存放位置在  $250_{(10)}$ ，每个元素占一个空间，问  $A[4][4]$  存放在什么位置？(脚注 $_{(10)}$ 表示用十进制表示)( )
  - A. 220
  - B. 250
  - C. 300
  - D. 350
5. 在双向循环链表中，在  $p$  指针所在节点后插入一个指针  $q$  所指向的新结点，修改指针的操作是( )
  - A.  $p->rlink = q; q->llink = p; p->rlink->llink = q; q->rlink = q;$
  - B.  $p->rlink = q; p->rlink->llink = q; q->llink = p; q->rlink = p->rlink;$
  - C.  $q->llink = p; q->rlink = p->rlink; p->rlink->llink = q; p->rlink = q;$
  - D.  $q->rlink = p->rlink; q->llink = p; p->rlink = q; p->rlink = q;$
6. 设森林 F 中有三棵树，第一，第二，第三棵树的节点个数分别为  $A_1, A_2$  和  $A_3$ ，则与森林 F 对应的二叉树根节点的左子树上的结点个数是( ).
  - A.  $A_1$
  - B.  $A_1 - 1$
  - C.  $A_2$
  - D.  $A_2 + A_3$
7. 若一棵二叉树的后序遍历序列是 {1,3,2,6,5,7,4}，中序遍历序列是 {1,2,3,4,5,6,7}，则下列哪句是错的？( )
  - A. 2 是 1 和 3 的父节点
  - B. 7 是 5 的父节点
  - C. 这是一棵二叉搜索树
  - D. 这是一棵完全二叉树
8. 一棵非空  $k$  ( $k \geq 2$ ) 叉树 T 中每个非叶子节点都有  $k$  个孩子，若 T 的高度为  $h$ (单节点

的树  $h = 1$ ), 则 T 的结点数最少为( )

A.  $\frac{k^{h-1}}{k} - 1$       B.  $k(h-1) + 1$       C.  $kh$       D.  $\frac{k^{h-1}-1}{k-1} + 1$

9. 一棵 124 个叶结点的完全二叉树, 最多有( )个结点.

A. 247      B. 248      C. 249      D. 250

10. 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( )个叶子节点.

A.  $\sum_{i=1}^m (i-1)N_i$       B.  $\sum_{i=1}^m N_i$       C.  $\sum_{i=2}^m N_i$       D.  $1 + \sum_{i=2}^m (i-1)N_i$

## 二、填空题(2 × 10)

1. 广义表 A=((a, b), (), (((), c)), d, e) 的长度为\_\_\_\_\_，深度为\_\_\_\_\_.
2. 设在一棵度数为 3 的树中, 度数为 3 的结点数有 3 个, 度数为 2 的结点数有 4 个, 度数为 1 的结点数有 5 个, 那么度数为 0 的结点数有\_\_\_\_\_个.
3. 假设一个顺序循环队列中有 X 个存储单元, 指针 F 指向队头元素的当前位置, 指针 R 指向队尾元素的后一个位置, 则判断队满的条件为\_\_\_\_\_, 队列中元素个数为\_\_\_\_\_.
4. 以二叉链表作为二叉树存储结构, 对有 K 个节点的二叉链表进行线索化时, 可以作为线索的链域的个数为\_\_\_\_\_. 若该二叉树左子树为空, 则在先序线索化后其空链域的个数为\_\_\_\_\_.
5. 设输入序列为 a, b, c, d, 则经过入栈和出栈的组合后可以得到\_\_\_\_\_种不同的输出序列.
6. 已知广义表 A = ((a, b, c), (d, e, f), (h, (i, j)), g), 通过 head 和 tail 运算从 A 表中取出原子项 e 的运算是\_\_\_\_\_.
7. 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 7, 13, 2, 6, 32, 3, 21, 10, 根据这些频率作为权值构造哈夫曼树, 则这颗哈夫曼树的高度为\_\_\_\_\_.

## 三、解答题(6 × 6)

1. 借助栈可以实现后缀表达式 A B C \* D / E - F G H \* + / - 的计算. 写出该计算中每一次栈操作后栈内的内容以及对应的栈操作, 并写出最后的中缀表达式.
2. 已知五个字符 S, T, U, X, Y 的哈夫曼编码分别是 000, 001, 01, 10, 11. 下列的 01 串是由以上 5 个字母构成的一段文本的哈夫曼编码:  
**111010000110100111100011001110111**  
(1) 请将上述 01 串还原为编码前的文本;  
(2) 假设字符 S, T, U, X, Y 的权值分别是 1, 2, 3, 4, 5, 请画出该哈夫曼树并计算带权路径长度WPL.

3. 设二叉树 T 的存储结构如下：

	0	1	2	3	4	5	6	7	8	9
Lchild	1	3	-1	6	-1	9	-1	-1	-1	-1
Data	A	B	C	D	E	F	G	H	I	J
Rchild	2	4	5	7	8	-1	-1	-1	-1	-1

其中 Lchild, Rchild 分别为结点的左, 右孩子指针域, Data 为结点的数据域.

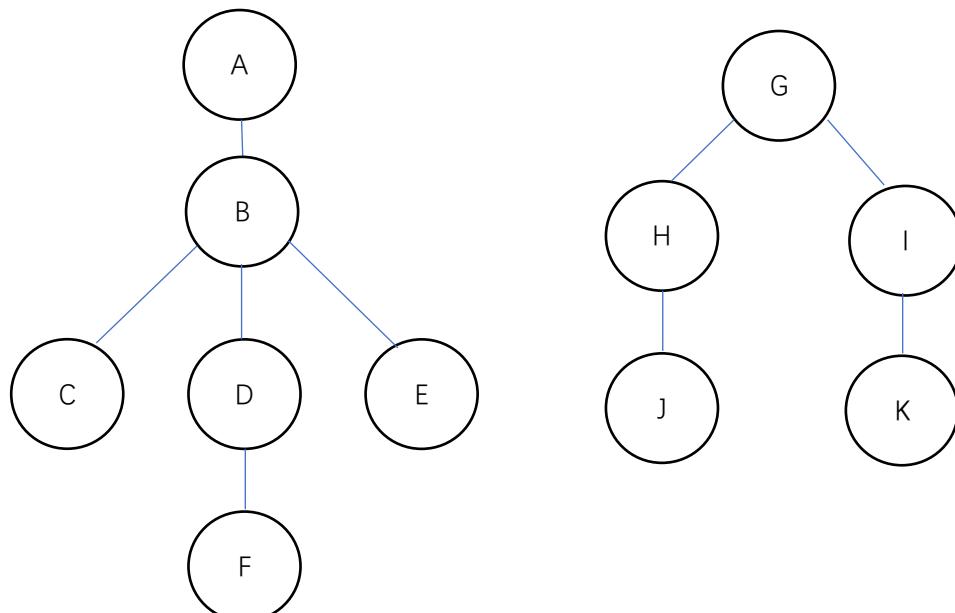
- (1) 画出二叉树 T 的逻辑结构;
- (2) 画出二叉树的中序线索化二叉树.

4. 在 KMP 算法中用字符串模式匹配时, 我们需要计算模式串的 Next 特征向量. 假设模式串  $P = "ABCABABCABC"$ , 求出对应的 Next 特征向量. Next 特征向量的对应数值表示: 当模式串 P 中第  $j$  个字符与目标串 T 中相应字符失配时, 模式 P 中应当由第  $\text{Next}[j]$  个字符与目标串中刚适配的字符重新继续进行比较.

下标 j	0	1	2	3	4	5	6	7	8	9	10
P	A	B	C	A	B	A	B	C	A	B	C
Next[j]	-1										

5. 最大堆初始为空, 然后依次插入元素 4, 6, 9, 8, 15, 10, 30, 以树的形式画出每插入一个元素并进行调整后堆的结果(共 7 棵树, 无需画出中间过程).

6. 给定下图所示的森林  $F = \{T_1, T_2\}$ , 基于子女-兄弟表示, 先将  $T_1$  与  $T_2$  分别转化为二叉树, 再将  $F$  转化为二叉树.



## 四、算法题 (24)

### 1. 算法填空(12分, 每空3分)

利用栈进行二叉树的中序遍历(非递归), 请在下划线部分填上正确的语句. 二叉树的结点(BinTreeNode)表示为:

leftChild	data	rightChild
-----------	------	------------

```
template<class T>
void BinaryTree<T>::InOrder(void(*visit)(BinTreeNode<T> *t)){
    Stack<BinTreeNode<T>> S;
    BinTreeNode<T> *p = root;
    do{
        while(p != NULL){
            _____ 1 _____;
            _____ 2 _____;
        }
        if(!S.IsEmpty()){
            S.Pop(p); visit(p);
            _____ 3 _____;
        }
    }while(_____ 4 _____);
};
```

- |          |
|----------|
| 1. _____ |
| 2. _____ |
| 3. _____ |
| 4. _____ |

### 2. 算法设计与实现: 装饰圣诞树(12)

小蓝鲸从商店内买回来一棵圣诞树, 其形式为一棵满二叉树, 每个节点上都放有一些小礼物. 由于礼物都是挂在树上的, 且每个树枝挂的礼物数目不同(每个礼物的重量为1, 每棵子树的重量为该子树内所有礼物重量之和), 导致圣诞树无法平衡. 为了保证圣诞树整体的平衡(注意每个子树也会不平衡), 小蓝鲸希望去商店再买一些礼物挂上. 由于预算有限, 小蓝鲸希望购买最少的礼物, 请你帮他计算需要购买的礼物数量.

请你设计相应的节点数据结构(2分), 写出算法思路计算小蓝鲸需要购买的最少的礼物(6分), 并写出相应的C++代码(4分).

Struct TreeNode /\*这里真的是Struct, 我不知道为什么, 我就是个敲卷子的苦力\*/

{

```
TreeNode * left;
TreeNode * right;

}/*这里真的没有分号，我不知道为什么，我就是个敲卷子的苦力*/
int balanceTree(TreeNode * root)
```