

尊敬的面试官您好：

我由于还在职，刚才面试完后忙了点日常工作内容，写代码的时间很紧，原谅我没有在代码中加入二叉树的定义和对代码的测试用例。。。但是我用自己构造的几个数据已经试验过了。。第一题求解正确了。

对于第二题，我写到最后出了点问题，我感觉我还是没理解懂“字符位置任意”这个话的含义吧，如果只是一个随意的字符串，那我会采用将字符串反转过来，然后把问题转换成求最长公共子字符串。

求二叉树每一层节点的和中最大的值

```
/*
 * 获取每一层的节点 value域的和
 */
private int sumNodeValuePerLevel(TreeNode pRoot, int level) {
    if (!pRoot || level < 0)
        return 0;
    if (0 == level) {
        return pRoot.getNodeValue();//获取节点的value域的值
    }
    return sumNodeValuePerLevel(pRoot.lchild, level - 1)
        + sumNodeValuePerLevel(pRoot.rchild, level - 1);
}

/*
 * 获取树的高度
 */
private int treeDepth(TreeNode pRoot) {
    if (null == pRoot) {
        return 0;
    }
    int left = treeDepth(pRoot.lchild);
    int right = treeDepth(pRoot.rchild);
    return left > right ? left + 1 : right + 1;
}

/*
 * 求层次节点域之和最大的一层的和数
 */
private int maxSumNodeValue(TreeNode pRoot) {
    if (!pRoot)
        return 0;
    int depth = treeDepth(pRoot);
    int maxSumValue = sumNodeValuePerLevel(pRoot, 0);
    for (int i = 1; i < depth; i++) {
        int sumValue = sumNodeValuePerLevel(pRoot, i);
        maxSumValue = (sumValue > maxSumValue) ? sumValue : maxSumValue;
    }
    return maxSumValue;
}
```

求任意字符串的最长回文子字符串

```
private int getMaxPlalindromeStringCount(String string) {
    if (string.isEmpty())
        return 0;
    int length = string.length();
    ArrayList<String> array = new ArrayList<String>(length);
    char[] chars = string.toCharArray();
    for (int i = 0; i < length; i++) {
        char ch = chars[i];
        int num = 0;
        for (int j = 0; j < length; j++) {
            if (ch == chars[j]) {
                num++;
            }
        }
    }
}
```

```
}  
if (num % 2 == 0) {  
    // 偶数  
    int k = 0;  
    while (num > 0) {  
        array.add(ch + "");  
        array.add(length - k, ch + "");  
        k++;  
        num = num - 2;  
    }  
} else {  
    // 奇数  
    while (num >= 0) {  
        array.add(num, ch + "");  
        num--;  
    }  
}  
}  
//出问题了  
}
```