

第四周习题刷题记录

226.翻转二叉树

题目：翻转一棵二叉树

算法：

1. 递归

算法：

1. 假设给出的字符集是ASCII字符，因此可以将ASCII码的范围0-127为数组下标，因此可以开辟一个128的数组。若出现一次将该下标对应的值设定为1，若再次出现则说明该字符串有相同的字符。

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     struct TreeNode *left;
 *     struct TreeNode *right;
 * };
 */

struct TreeNode* invertTree(struct TreeNode* root){
    if(NULL == root) {
        return NULL;
    }
    struct TreeNode *right = invertTree(root->right);
    struct TreeNode *left = invertTree(root->left);
    root->left = right;
    root->right = left;
    return root;
}
```

165.比较版本号

题目：比较两个版本号 version1 和 version2。

如果 version1 > version2 返回 1，如果 version1 < version2 返回 -1，除此之外返回 0。

算法：

- 1.解析字符串，逐个比较两个字符串对应位置处的数字的大小

```
/**
 * @lc app=leetcode.cn id=165 lang=c
 *
 * [165] 比较版本号
 */

// @lc code=start

int compareVersion(char * version1, char * version2){
    int n = strlen(version1);
```

```
int m = strlen(version2);
int i = 0;
int j = 0;
while (i < n || j < m) {
    int x = i;
    int y = j;
    int sum1 = 0;
    int sum2 = 0;
    while (x < n && version1[x] != '.') {
        sum1 = sum1 * 10 + version1[x++] - '0';
    }
    while (y < m && version2[y] != '.') {
        sum2 = sum2 * 10 + version2[y++] - '0';
    }
    if (sum1 > sum2) {
        return 1;
    } else if (sum1 < sum2) {
        return -1;
    }
    i = x + 1;
    j = y + 1;
}
return 0;
}
```