

## 剑指26 树的子结构

不会。。。

但是说到递归提示就会做了。

遇到树的问题，往往设计递归和树的遍历。

解题思路可为：遍历A的每个节点，判断以当前节点为root的A的子树和B树之间是否有同root子结构关系。

```
1  /**
2   * Definition for a binary tree node.
3   * struct TreeNode {
4   *     int val;
5   *     TreeNode *left;
6   *     TreeNode *right;
7   *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
8   * };
9   */
10 class Solution {
11 public:
12     bool compare(TreeNode* A, TreeNode* B) {
13         if (B != NULL && A == NULL) return false;
14         else if (B == NULL) {
15             return true;
16         }
17         else if (A->val != B->val){
18             return false;
19         }
20         return compare(A->right, B->right) && compare(A->left, B->left);
21     }
22
23
24     bool isSubStructure(TreeNode* A, TreeNode* B) {
25         if (A == NULL || B == NULL) return false;
26
27         if (A->val == B->val && compare(A, B)) {
28             return true;
29         }
30
31         return isSubStructure(A->left, B) || isSubStructure(A->right, B);
32     }
33 };
```