

剑指Offer（六十一）：序列化二叉树

🕒 2018年1月30日 13:13:49 📄 2 🌡 4,304 °C 🛠 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

请实现两个函数，分别用来序列化和反序列化二叉树。

1、思路

这道题思路简单，使用前序遍历来序列化和反序列化即可。只要自己写的程序格式对应上即可。可以使用\$符号表示NULL，同时每个结点之间，需要添加逗号，即','进行分隔。

直接看代码即可。

2、代码

C+

```
1  /*
2  struct TreeNode {
3      int val;
4      struct TreeNode *left;
5      struct TreeNode *right;
6      TreeNode(int x) :
7          val(x), left(NULL), right(NULL) {
8      }
9  };
10 */
11 class Solution {
12 public:
13     char* Serialize(TreeNode *root) {
14         if(!root){
15             return NULL;
16         }
17         string str;
18         SerializeCore(root, str);
19         // 把str流中转换为字符串返回
20         int length = str.length();
21         char* res = new char[length+1];
22         // 把str流中转换为字符串返回
23         for(int i = 0; i < length; i++){
24             res[i] = str[i];
25         }
26         res[length] = '\0';
27         return res;
28     }
29     TreeNode* Deserialize(char *str) {
30         if(!str){
31             return NULL;
32         }
33         TreeNode* res = DeserializeCore(&str);
34         return res;
35     }
36     void SerializeCore(TreeNode* root, string& str){
37         // 如果指针为空，表示左子节点或右子节点为空，则在序列中用#表示
38         if(!root){
39             str += '#';
40             return;
```

```
41     }
42     string tmp = to_string(root->val);
43     str += tmp;
44     // 加逗号, 用于区分每个结点
45     str += ',';
46     SerializeCore(root->left, str);
47     SerializeCore(root->right, str);
48 }
49 // 递归时改变了str值使其指向后面的序列, 因此要声明为char**
50 TreeNode* DeserializeCore(char** str){
51     // 到达叶节点时, 调用两次, 都返回null, 所以构建完毕, 返回父节点的构建
52     if(**str == '#'){
53         (*str)++;
54         return NULL;
55     }
56     // 因为整数是用字符串表示, 一个字符表示一位, 先进行转换
57     int num = 0;
58     while(**str != ',' && **str != '\0'){
59         num = num * 10 + ((*str) - '0');
60         (*str)++;
61     }
62     TreeNode* root = new TreeNode(num);
63     if(**str == '\0'){
64         return root;
65     }
66     else{
```



微信公众号

分享技术, 乐享生活: 微信公众号搜索

「JackCui-AI」关注一个在互联网摸爬滚  
打的潜行者。

知之者不如好之者, 好之者不如乐之者。--- 孔子