面试题 37：序列化二叉树

题目：请实现两个函数，分别用来序列化和反序列化二叉树。

解

用前序遍历

```
char* serialize(TreeNode* root)
{
    if(!root)
        return nullptr;
    string str;
    help(root,str);
    char *res=new char[str.size()+1];
    int i;
    for(i=0;i<str.size();i++)
        res[i]=str[i];
    res[i]='\0';
    return res;
}
void help(TreeNode* node,string &str)
{
    if(!node)
    {
        str+='#';
        return;
    }
    string s=to_string(node->val);
    str+=s;
    str+='_';
    help(node->left,str);
    help(node->right,str);
}
```

反序列化

```
TreeNode* deserialize(char *str)
{
    if(str==nullptr)
        return nullptr;
    TreeNode *res=help(&str);
    return res;
}
```

```
TreeNode* help(char **str)
{
    if(**str=='#')
    {
        (*str)++;
        return nullptr;
    }
    int num=0;
    while(**str!='_' && **str!='\0')
    {
        num=num*10+((**str)-'0');
        (*str)++;
    }
    TreeNode* root=new TreeNode(num);
    if(**str== '\0')
    {
        return root;
    }
    else
        (*str)++;
    root->left=help(str);
    root->right=help(str);

    return root;
}
```

leetcode版本

```
class Codec {
public:

    // Encodes a tree to a single string.
    string serialize(TreeNode* root) {
        ostringstream out;
        se(root,out);
        return out.str();
    }

    // Decodes your encoded data to tree.
    TreeNode* deserialize(string data) {
        istringstream in(data);
        return de(in);
    }
private:
    void se(TreeNode* root,ostringstream &out)
    {
        if(root)
```

```
        {
            out<<root->val<<' ';
            se(root->left,out);
            se(root->right,out);
        }
        else
            out<<"# ";
    }
    TreeNode* de(istringstream &in)
    {
        string val;
        in>>val;
        if(val=="#")
            return nullptr;
        TreeNode* root=new TreeNode(stoi(val));
        root->left=de(in);
        root->right=de(in);
        return root;
    }
};
```