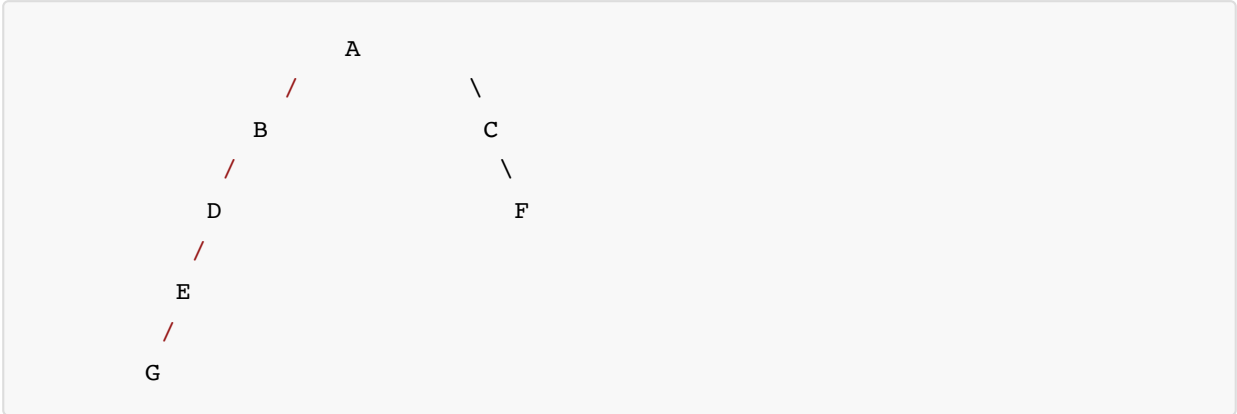
20180329_Akuna 公司的CPP笔试题目

题目大概是给出如下输入：

```
"(B,D) (D,E) (A,B) (C,F) (E,G) (A,C)"
```

上面是一个字符串，需要自己解开。

实际是表述了这样一颗树。



需要检查这么几种错误:

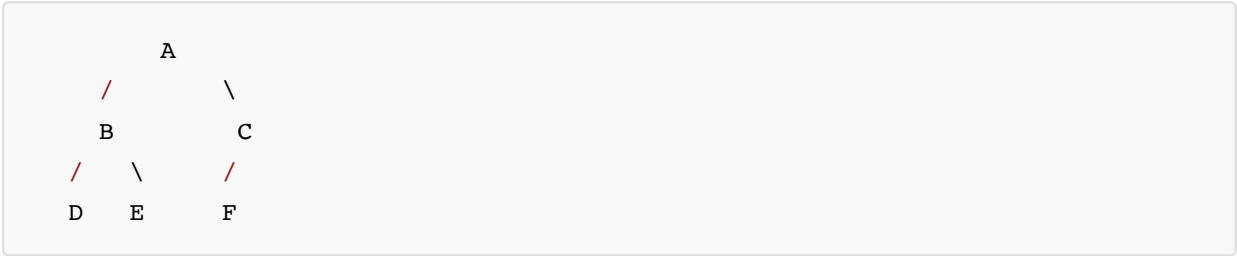
1- 一个节点有多个父节点

2- 一个父节点有超过两个的子节点

3- 一个节点可能构成一个环

4- 其它错误

对于这样一颗树



需要输出的格式如下:【这种格式好像有个名字的】

```
(A(B(D)(E))(C(F)))
```

注意，左子节点要求 < 右子节点

程序如下：

```
#include <iostream>
#include <bits/stdc++.h>

using namespace std;


struct Node{
    string val;
    Node* left;
    Node* right;
    Node(string val):val(val),left(NULL),right(NULL){}
};

int getNode(string& nodes, int start, int& next,string& first, string&
second ){
    int i = start;
    while(nodes[i] != '(' && nodes[i] != '\0'  )
        i++; // cut the blank

    if(nodes[i]  == '\0'){
        return -2; // empty
    }

    if( i+4 < nodes.size() && nodes[i+1] >= 'A' && nodes[i+1] <= 'Z' &&
nodes[i+2] == ',' && ( (nodes[i+3] >= 'A' && nodes[i+3] <= 'Z') ||
(nodes[i+3] == ' ' ) ) && nodes[i+4] == ')' ){
        first = nodes[i+1];
        second = nodes[i+3];
        next = i + 5;
        return 0;
    }else{
        cout << "E5";       /// not complete nodes
        return -1;
    }

//    return -1;
}

string SExpression(string nodes) {
    /// process expetion: special input
    /// ....


    int next = 0;
    string first,second;
    std::unordered_map<string,Node* > mymap;
    for( int i =0; i< nodes.size(); ){
```

```
        int flag = getNode(nodes, i, next,first, second );

        if( flag == 0 ){
            Node* node = new Node(first);
            Node* node2 = new Node(second);
            if( auto it = mymap.find( second ) == map.end() ){  /// have no
this node
                mymap.insert(make_pair(second,node2));
            }

            if( auto it = mymap.find( first ) == map.end() ){
                node->left = node2;
                mymap.insert(make_pair(first,node));
            }else{
                if( it->second -> left != NULL ){
                    if( it->second -> left > second ){
                        it -> second -> right = it ->second -> left;
                        it -> second -> left = node2;
                    }else{
                        it -> second -> left = node2;
                    }
                }
            }


            i = next;
        }else if( flag == -2 ){
            break;
        }else if( flag == -1 ){
            break;
        }
    }

    Node* root = NULL;
    //// generate tree, 需要用到数组来记录，有一个节点没有出现在根节点的就是root节
点。
    for( auto it: mymap ){
        if(root == NULL){
            root = it.second;
        }

    }


}



int main()
```

```
{

   string nn = "(B,D) (D,E) (A,B) (C,F) (E,G) (A,C)";
    string res = SExpression(nn);
    cout << res; << endl;
    return 0;
}
```