```
class TreeNode{

private:
    TreeNode * parent; // 父节点的指针
    vector<TreeNode *> children; // 子节点的指针
    int id;              // 节点id
    int label;           //
    int pairId;

public:
    // TreeNode的构造函数与析构函数
    TreeNode();
    TreeNode(int id);
    TreeNode(int id,int label);
    TreeNode(int id,int label,int pairId);
    TreeNode(TreeNode *p);
    ~TreeNode();

    void AddChild(TreeNode *pChild);
    void SetId(int x) {id = x;}
    void SetLabel(const int str) {label = str;}
    void SetPairId(const int id) { pairId = id;}
    bool IsLeaf() const { return children.size() == 0;}
    bool IsRtLeaf() const { return pairId != -1;}
    bool Is2ndRtLeaf();
    bool IsRoot() const { return parent == NULL;}
    bool IsSibling(const TreeNode * p) const ;
    void removeChild(TreeNode * p);
    int GetChildrenSize() { return (int)children.size();}
    TreeNode * GetChild(int i) { return children[i];}
    TreeNode * GetParent() { return parent;}
    TreeNode * GetSiblingNode();
    TreeNode * GetRoot();
    int GetId() { return id;}
    int GetLablel() { return label;}
    int GetPairId() { return pairId;}
    string ToString();

    TreeNode * Clone(); // 克隆以当前的为根的子树
};
```