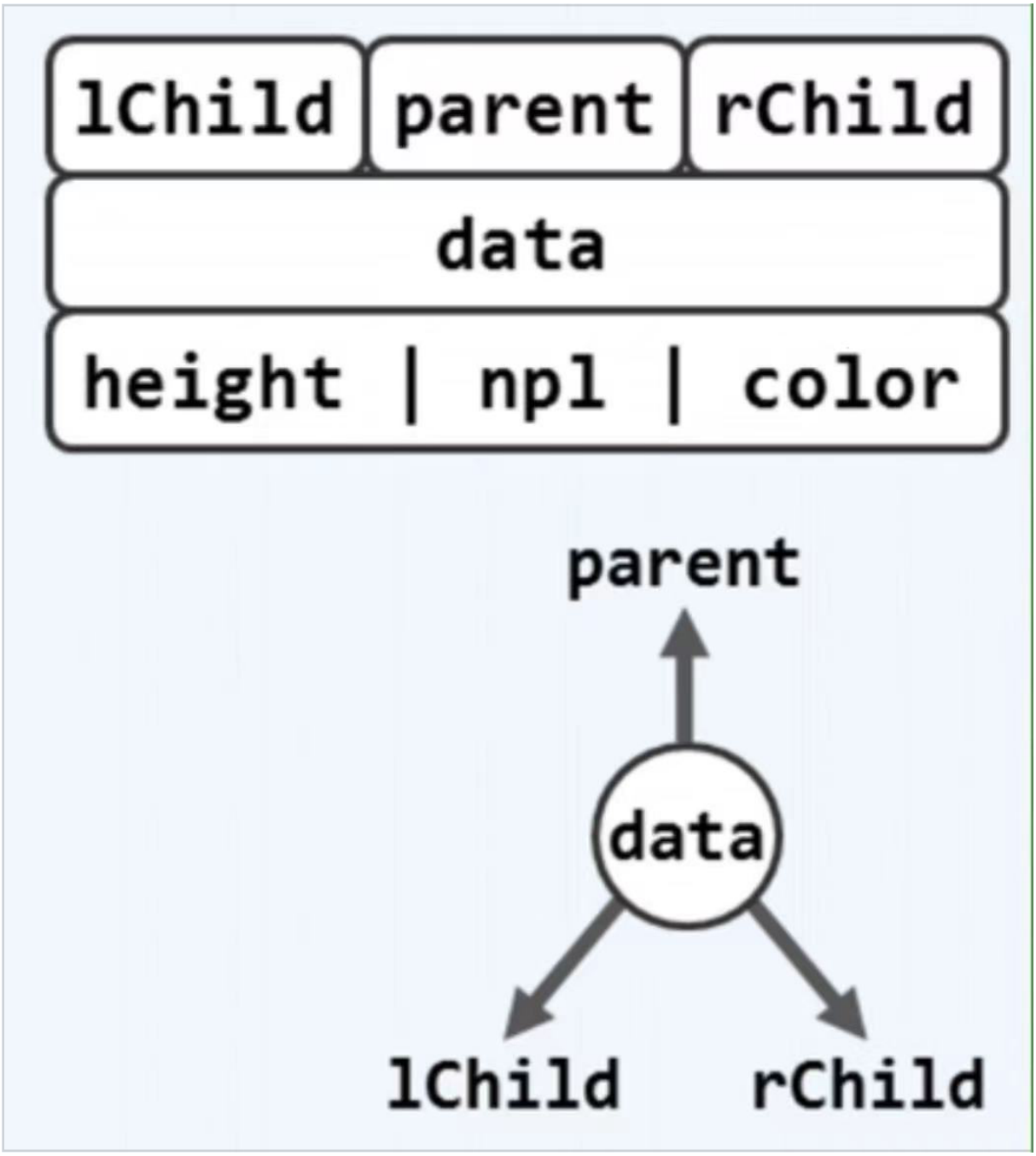
05-D 二叉树实现

#数据结构邓神

binNode模版类

数据组成：

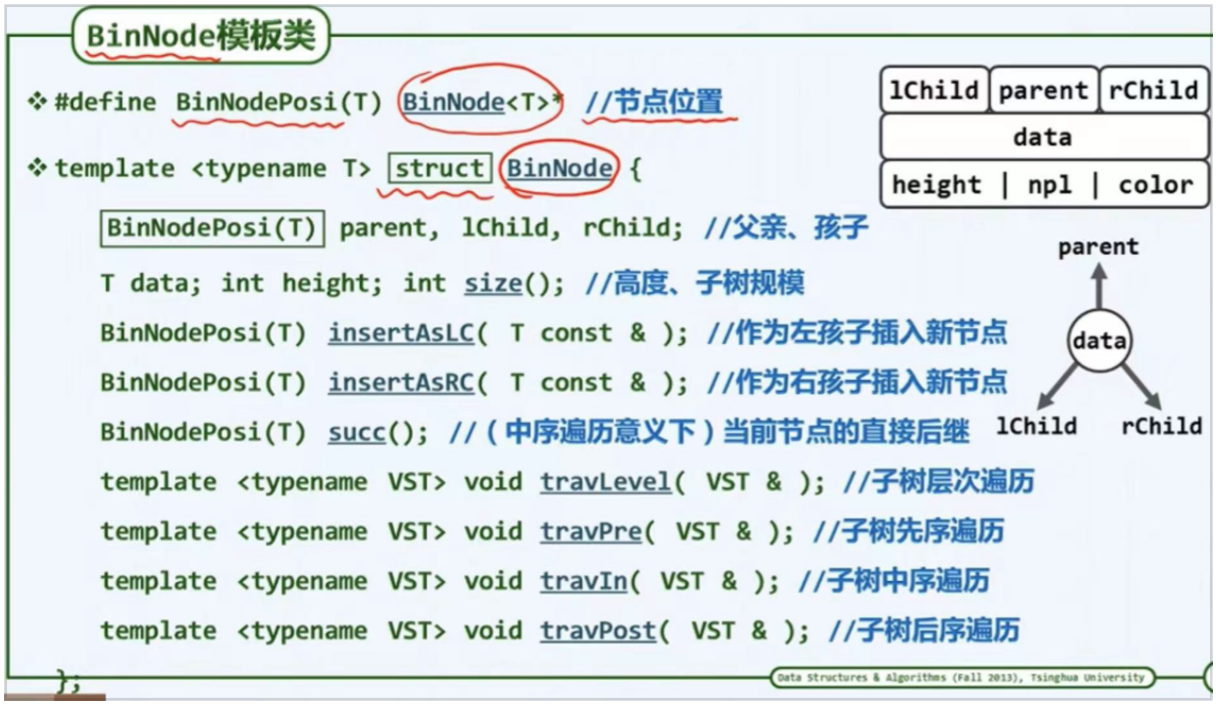


Height:高度

Color： 红黑树

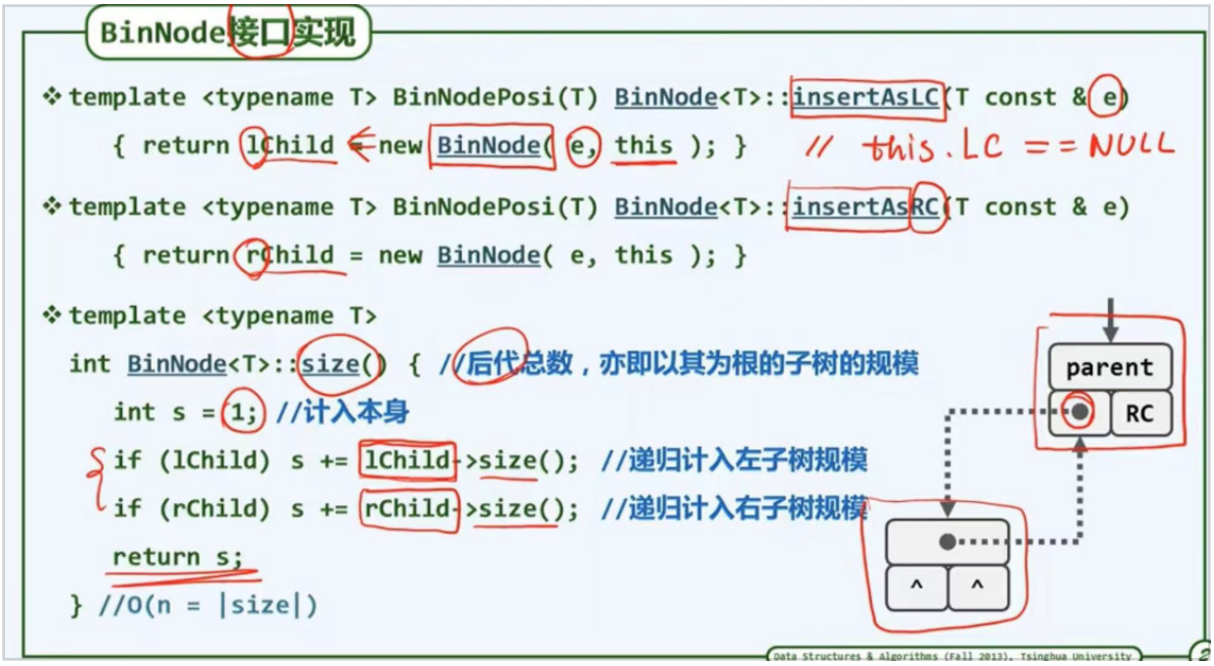
npl????

每一个节点都是通过引用去指向其他的节点，每一个节点也都是被其他的节点用引用指向



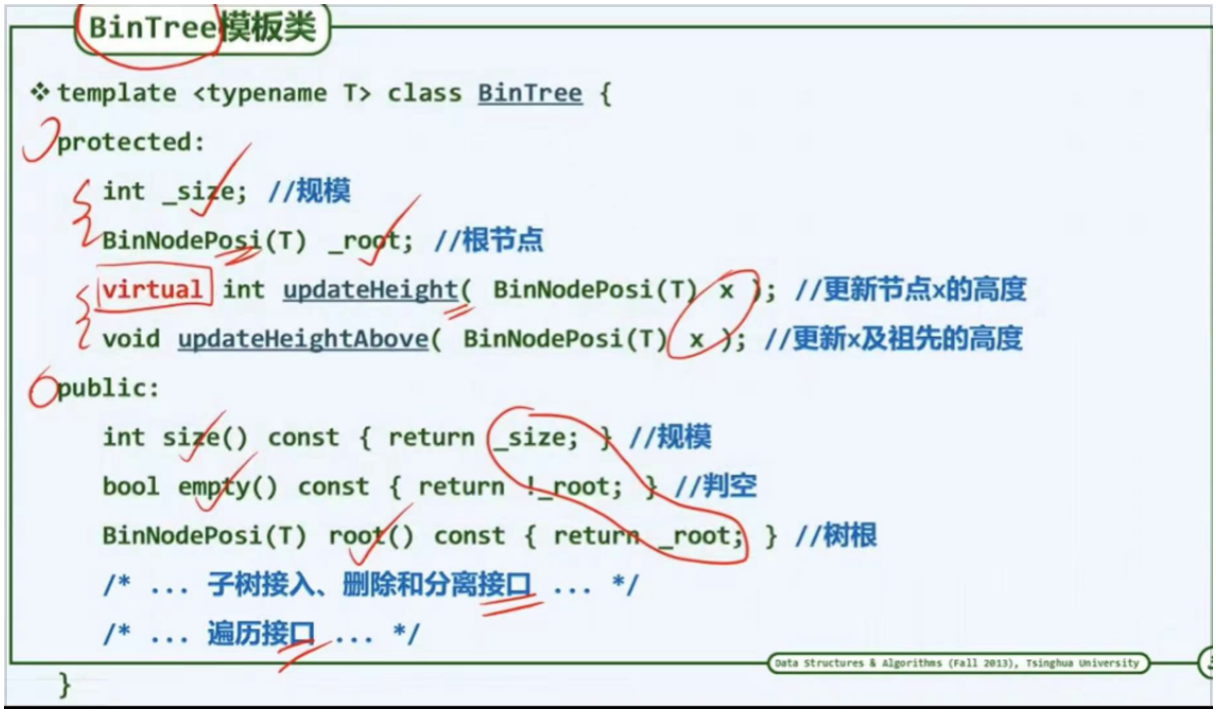

```cpp
#define BinNodePosi(T) BinNode<T>* // 将节点引号 直接变成 位置
template <typename T> struct BinNode {
    // 父亲和左右孩子 引用
    BinNodePosi(T) parent,lChild,rChild;
    // 数据域
    T data;
    // 高度和子树的规模
    int height;
    int size(); // 所有后代的总数
    BinNodePosi(T) insertAsLC(T const &);
    BinNodePosi(T) insertAsRC(T const &);
    BinNodePosi(T) succ();
    // 遍历
    template <typename VST> void travLevel(VST &); // 层次遍历
    template <typename VST> void travPre(VST &); // 子树先序遍历
    template <typename VST> void travIn(VST &); // 子树中序遍历
    template <typename VST> void travPost(VST &); // 子树后序遍历
};
```

binNode 接口实现



BinTree 模版类

```
// BinTree
template <typename T> class BinTree {
protected:
    int _size;
    BinNodePosi(T)_root;
    virtual int updateHeight(BinNodePosi(T)x);
    void updateHeightAbove(BinNodePosi(T)x);
public:
    int size() const { // 规模
        return _size;
    }
    bool empty() const { // 判空
        return !_root;
    }
    BinNodePosi(T)root() const { // 树根
        return _root;
    }
    /* 子树接入，删除和分离接口 */
    /* 遍历接口 ... */
};
```



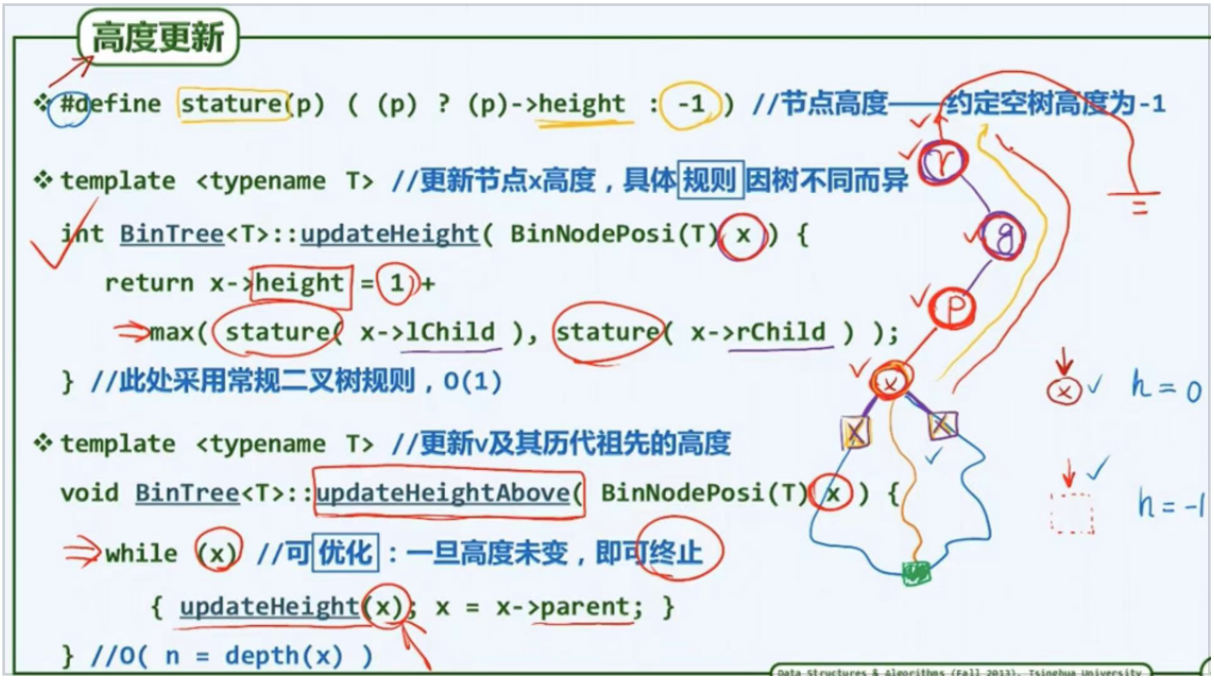
高度更新

高度就是以他为根的子树从他通往最深的叶节点的路径长度
如果只有单节点高度取为 0
不存在任何节点的树高度取为-1

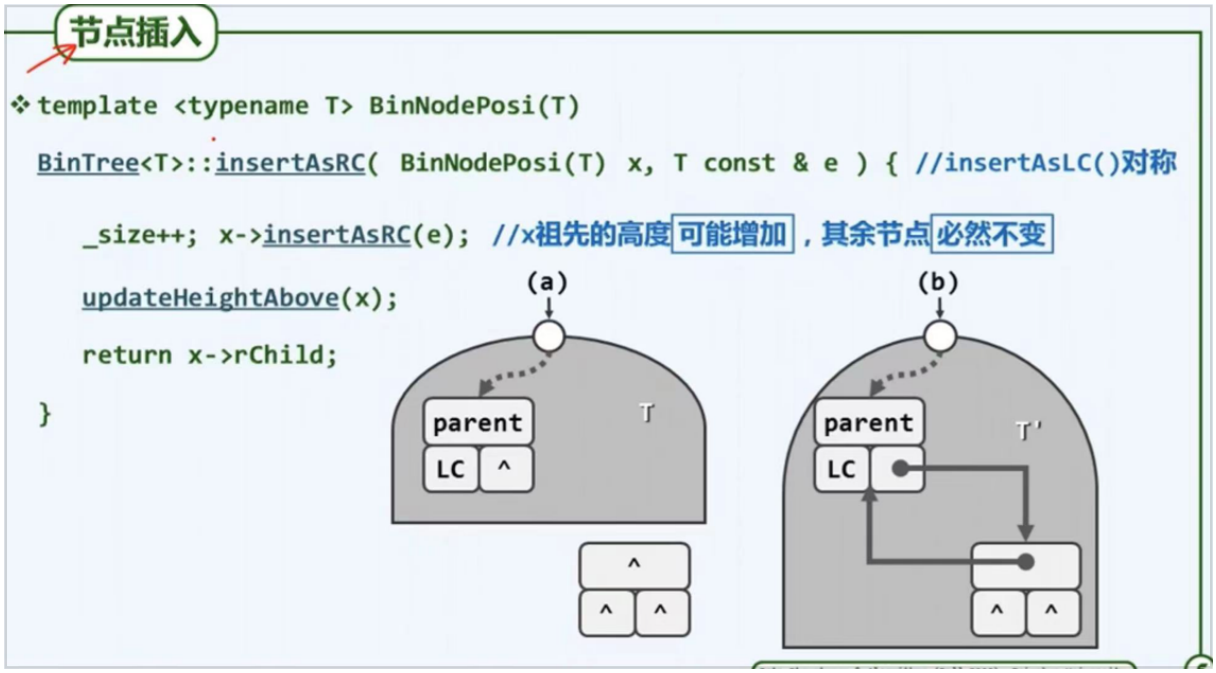
```
#define stature(p) ((p)?(p)->height : -1)
```

一个节点的高度应该等于他的左孩子或者右孩子的高度的更大者的高度再加一 （能否用递归算法求出？）

```
#define stature(p) ((p)?(p)->height : -1)
template <typename T> int BinTree<T>::updateHeight(BinNodePosi(T) x){
    return x->height = 1 + max(stature(x->lChild), stature(x->rChild));
} // 采取常规二叉树规则 O(1)
```



节点插入




```
// 插入节点
template <typename T> BinNodePosi(T) BinTree<T>::insertAsRC(BinNodePosi(T) x,T
const & e){
    _size++;
    x->insertAsRC(e);
    updateHeightAbove(x); // 在增加或者删除任何子节点都需要更新高度
    return x->rChild;
}
```

```
template <typename T> BinNodePosi(T) BinTree<T>::insertAsLC(BinNodePosi(T) x,T const & e){
    _size++;
    x->insertAsLC(e); // 这边这个insertAsLC 是BinNode提供的接口，不是这个BinTree的接口，即不是递归，这边也没有递归的意义
    updateHeightAbove(x);
    return x->lChild;
}
```