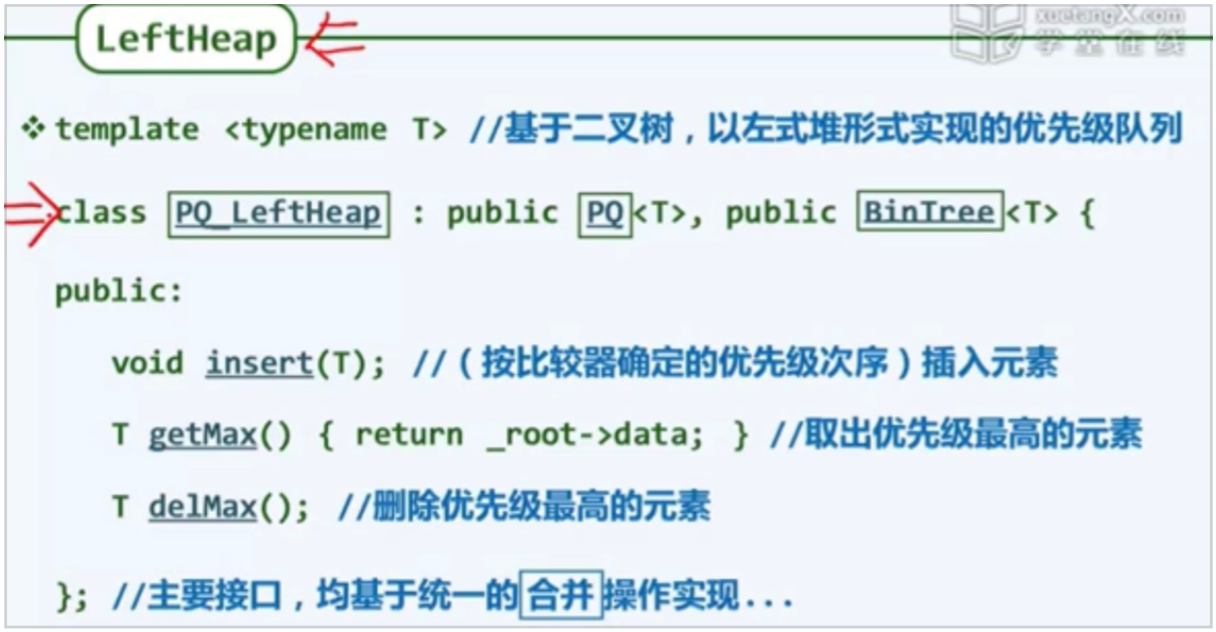
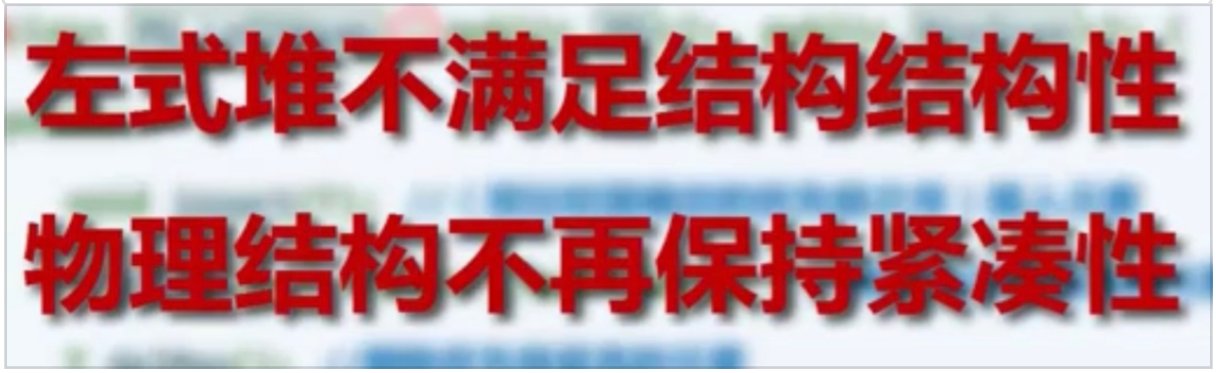
12F3 合并算法

#数据结构邓神

LeftHeap模版类

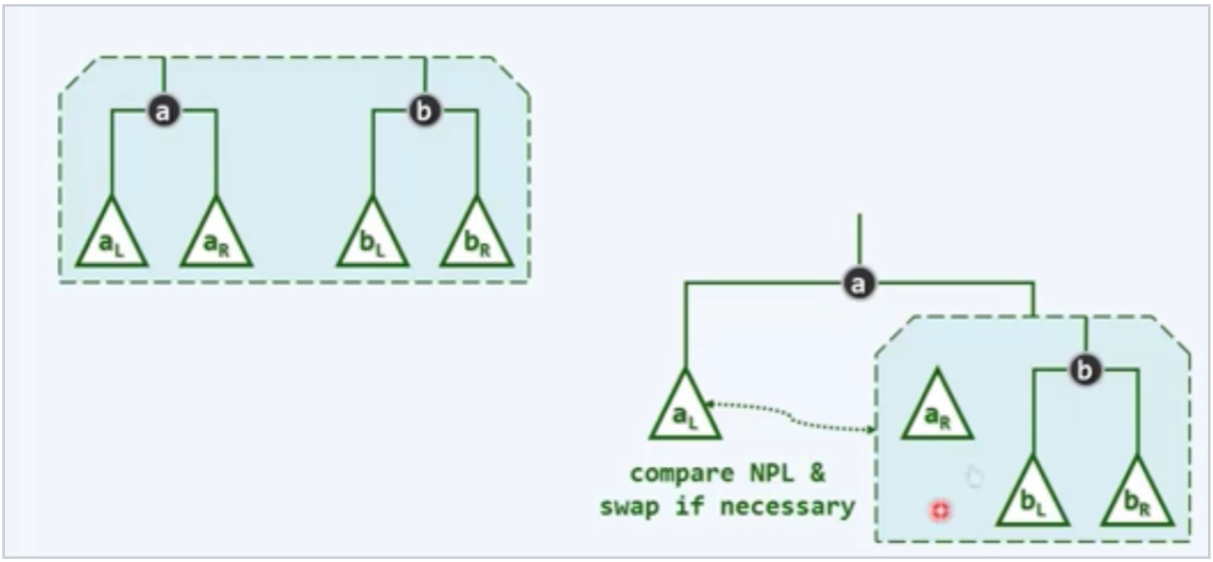



```
template <typename T> class PQ_LeftHeap : public PQ<T> , public BinTree <T> {
public:
    void insert(T);
    T getMax(){
        return _root->data;
    }
    T delMax();
};
```



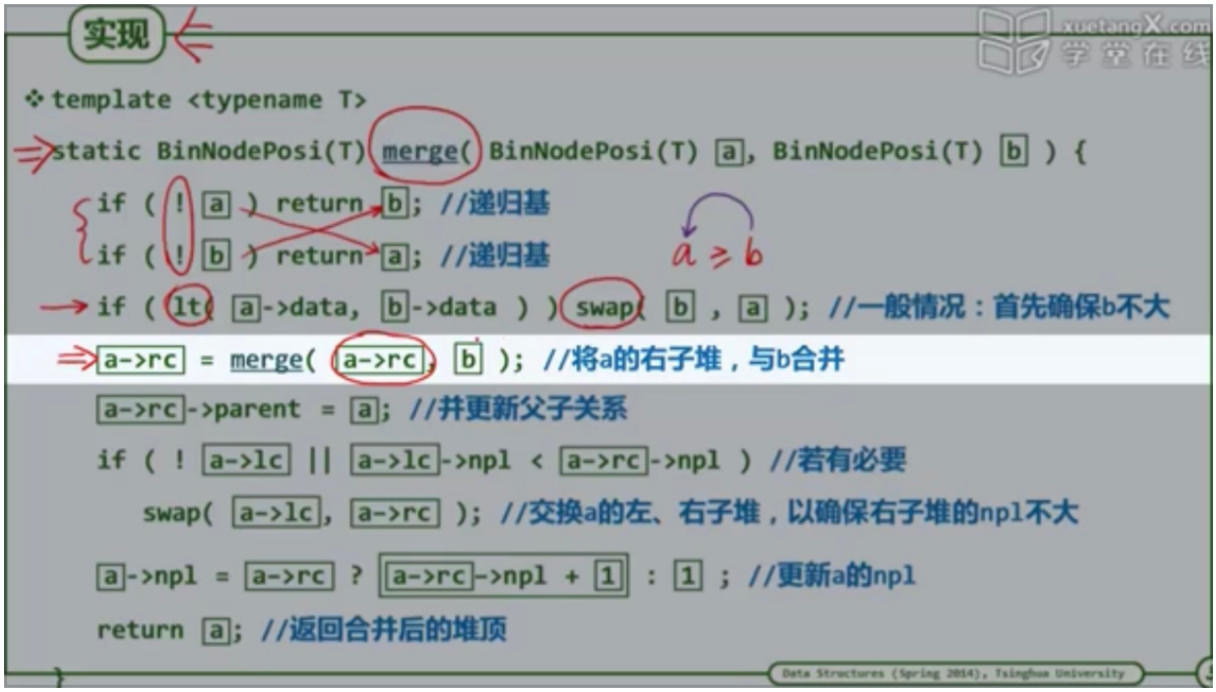

```
template <typename T>
static BinNodePosi(T) merge(BinNodePosi(T),BinNodePosi(T));
// 通过外部函数的形式给出具体算法
```

算法



在合并后如果NPL逆序就互换左右子树

实现




```
template <typename T> static BinNodePosi(T)
merge(BinNodePosi(T),BinNodePosi(T)){
    if (! a) {
        return b;
    }
    if (! b) {
        return a;
    }
```

```
    if ( a->data < b->data ){
        swap(b,a);
    }
    a->rc = merge(a->rc,b);
    a->rc->parent = a;
    if (! a->lc || a->lc->npl < a->rc->npl){
        swap(a->lc,a->rc);
    }
    a->npl = a->rc ? a->rc->npl + 1 : 1;
    return a;
}
```