10.优先级队列

（xa2）左式堆：合并

邓俊辉

deng@tsinghua.edu.cn

LeftHeap

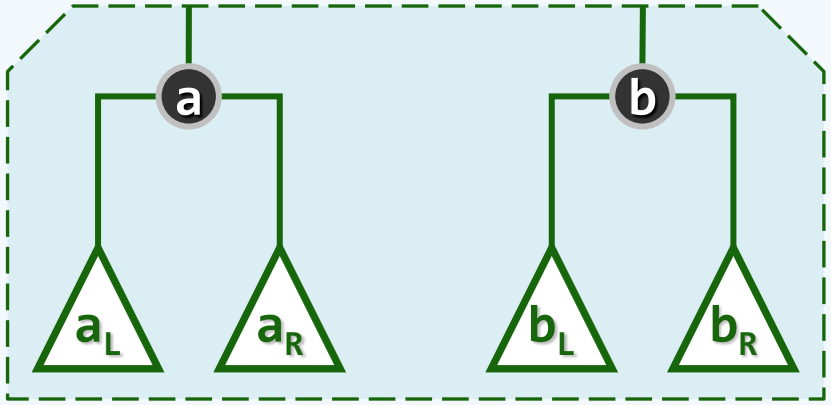
❖ template <typename T> //基于二叉树，以左式堆形式实现的优先级队列

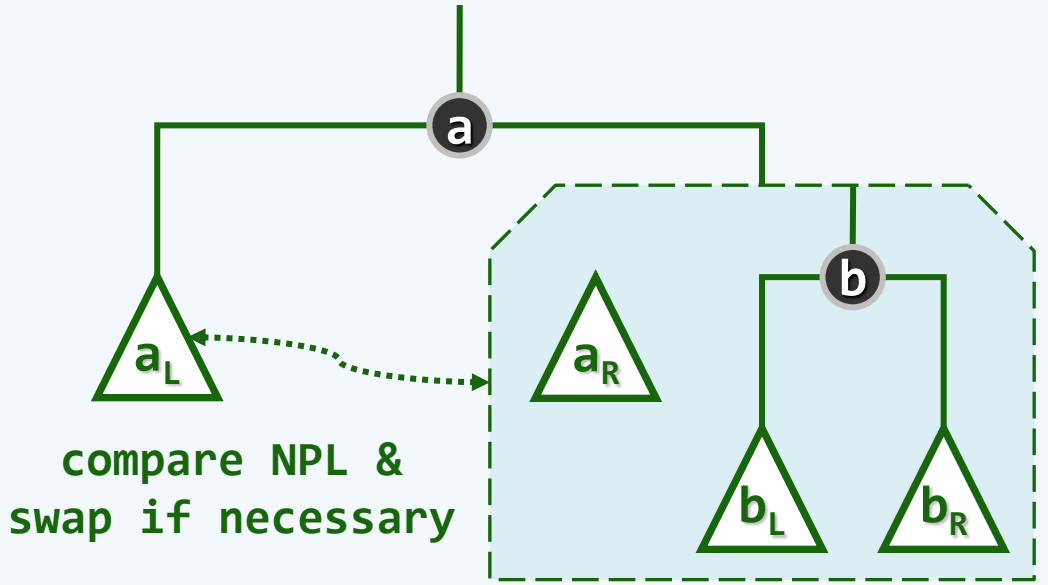
```
class PQ_LeftHeap : public PQ<T>, public BinTree<T> {
public:
    void insert(T); //（按比较器确定的优先级次序）插入元素
    T getMax() { return _root->data; } //取出优先级最高的元素
    T delMax(); //删除优先级最高的元素
}; //主要接口，均基于统一的合并操作实现...
```

❖ template <typename T>

```
static BinNodePosi(T) merge( BinNodePosi(T), BinNodePosi(T) );
```

算法

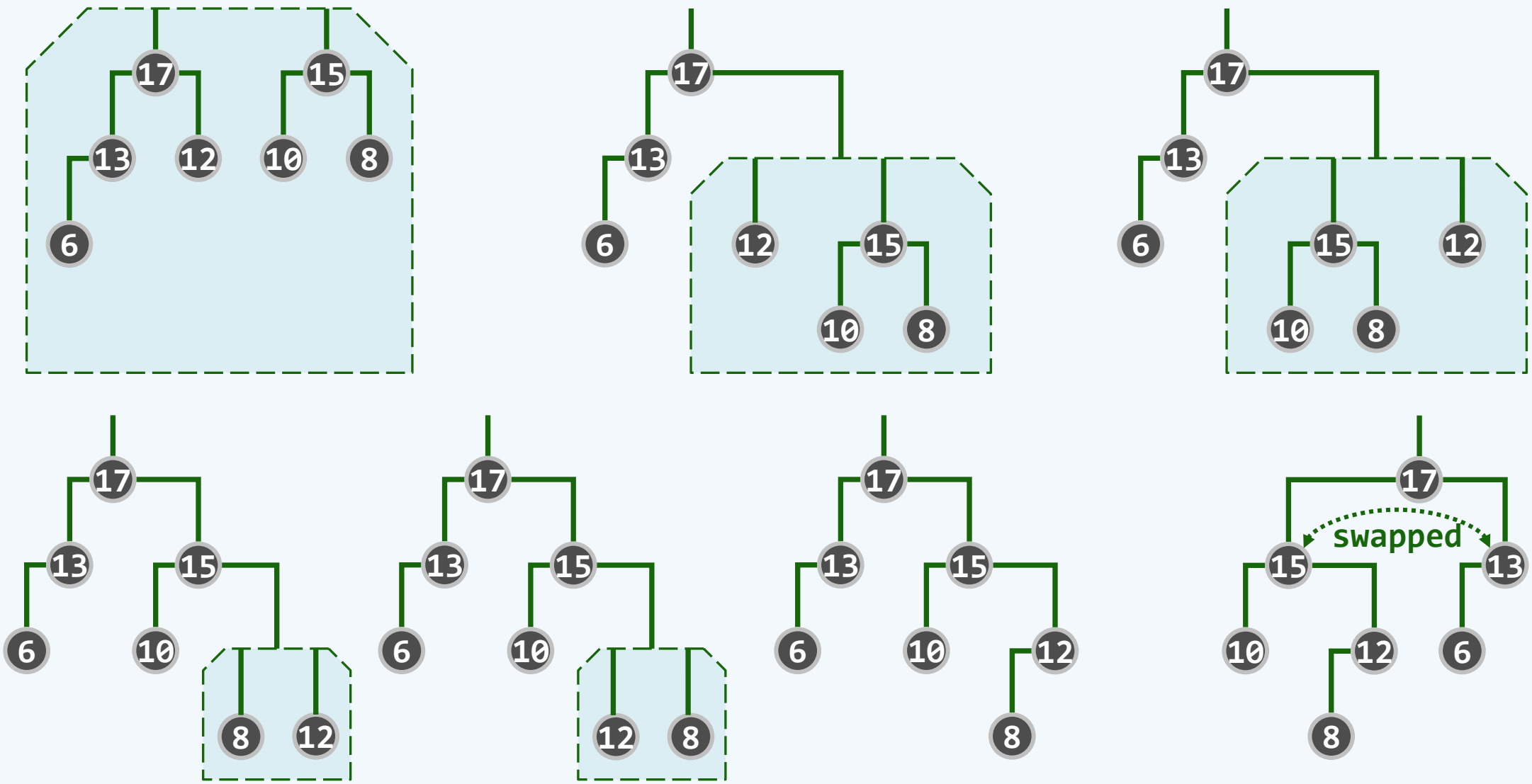




实现

```
❖ template <typename T>
  static BinNodePosi(T) merge( BinNodePosi(T) a, BinNodePosi(T) b ) {
     if ( ! a ) return b; //递归基
     if ( ! b ) return a; //递归基
     if ( lt( a->data, b->data ) ) swap( b , a ); //一般情况：首先确保b不大
     a->rc = merge( a->rc, b ); //将a的右子堆，与b合并
     a->rc->parent = a; //并更新父子关系
     if ( ! a->lc || a->lc->npl < a->rc->npl ) //若有必要
        swap( a->lc, a->rc ); //交换a的左、右子堆，以确保右子堆的npl不大
     a->npl = a->rc ? a->rc->npl + 1 : 1 ; //更新a的npl
     return a; //返回合并后的堆顶
  }
```

实例


17
15
13
12
10
8
6
17
13
6
12
15
10
8
17
13
6
15
12
10
8
17
13
15
6
10
8
12
17
13
15
6
10
12
8
17
13
15
6
10
12
8
17
swapped
15
13
10
12
6
8