10.优先级队列

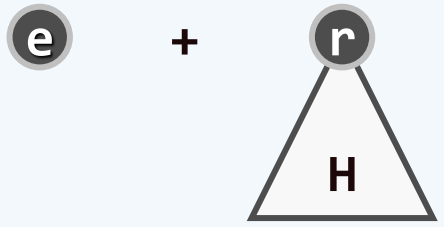
（xa3）左式堆：插入与删除

邓俊辉

deng@tsinghua.edu.cn

insert()

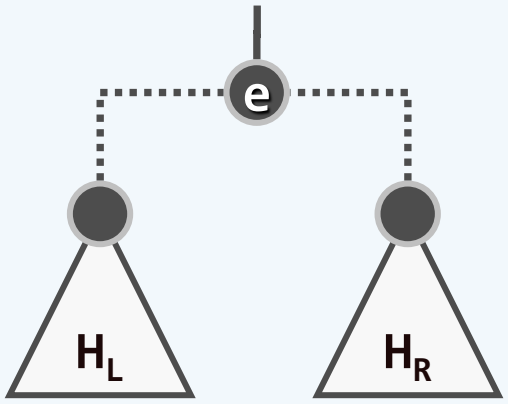
❖ template <typename T>




```
void PQ_LeftHeap<T>::insert( T e ) { //O(logn)
    BinNodePosi(T) v = new BinNode<T>( e ); //为e创建一个二叉树节点
    _root = merge( _root, v ); //通过合并完成新节点的插入
    _root->parent = NULL; //既然此时堆非空，还需相应设置父子链接
    _size++; //更新规模
}
```

delMax()

```
❖ template <typename T> T PQ_LeftHeap<T>::delMax() { //O(logn)
    BinNodePosi(T) lHeap = _root->lc; //左子堆
    BinNodePosi(T) rHeap = _root->rc; //右子堆
    T e = _root->data; //备份堆顶处的最大元素
    delete _root; _size--; //删除根节点
    _root = merge( lHeap, rHeap ); //原左、右子堆合并
    if ( _root ) _root->parent = NULL; //更新父子链接
    return e; //返回原根节点的数据项
}
```

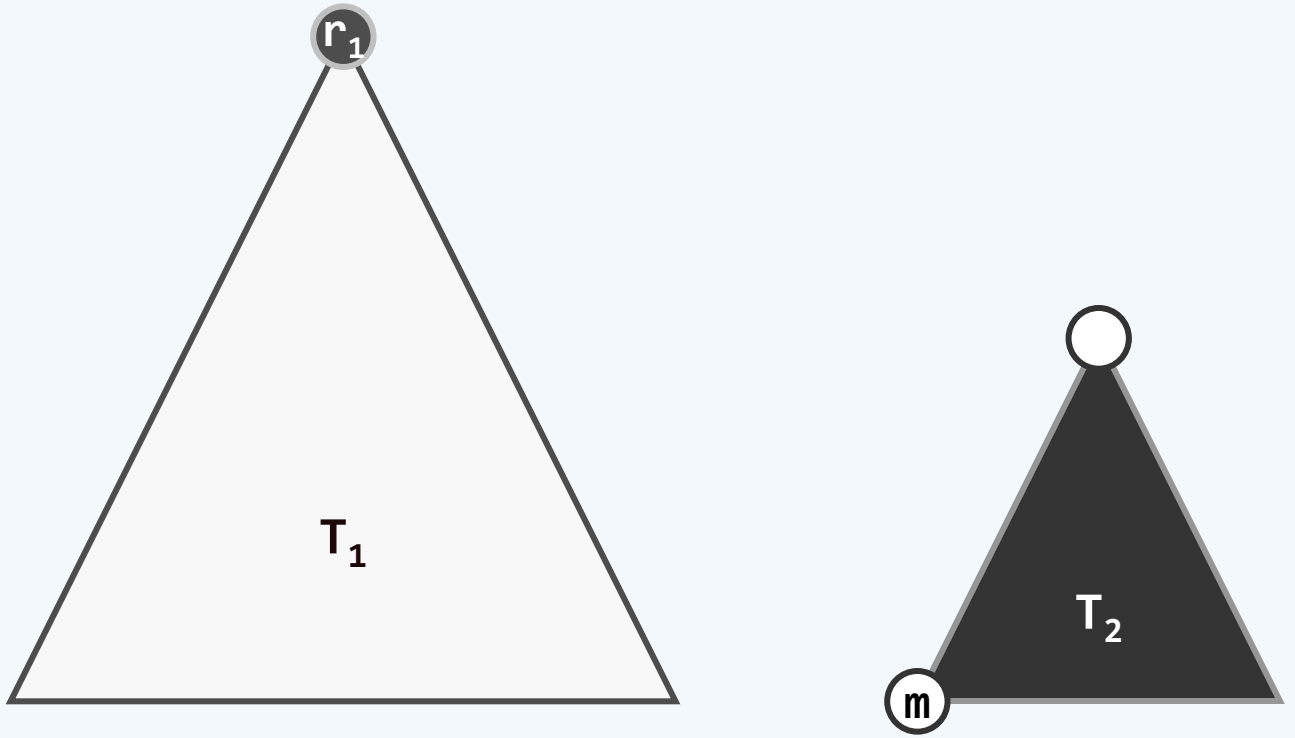


AVL::merge()

❖ 设T_1和T_2为两棵AVL树，且 $\max(T_1) < m = \min(T_2)$

如何**尽快**地将其合并为一棵AVL树？

❖ WLOG， $\text{height}(T_1) \geq \text{height}(T_2)$



AVL::merge()