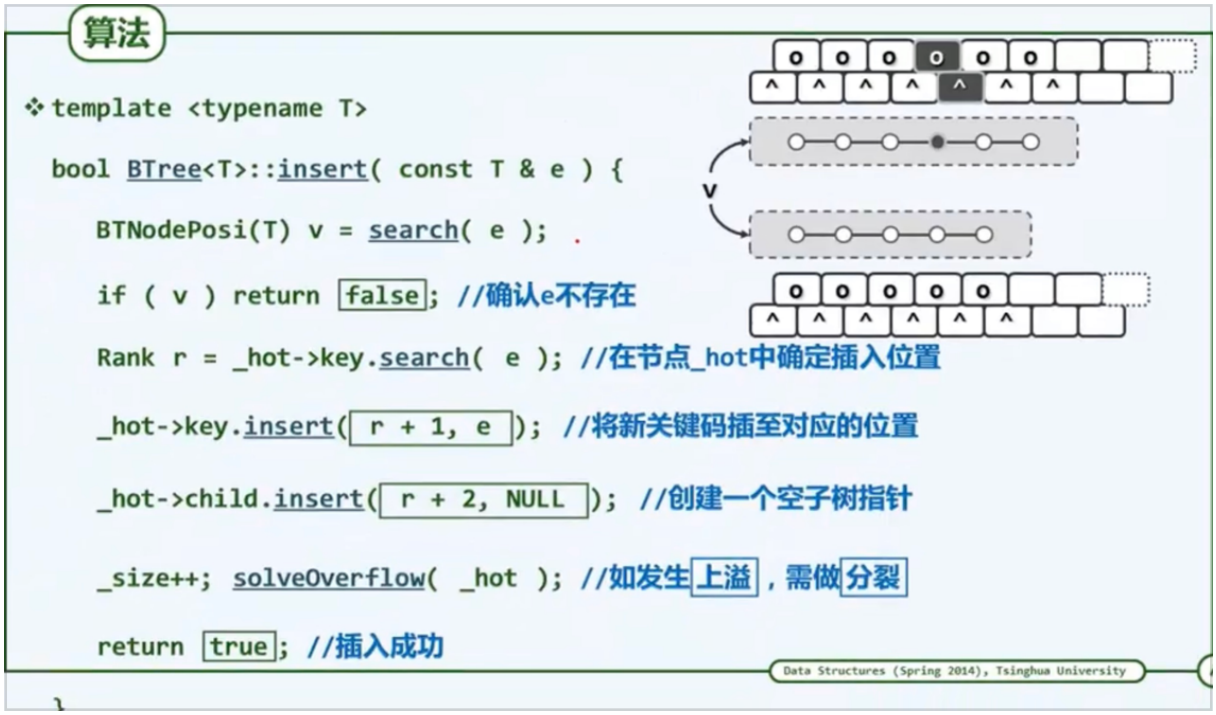
10B4 插入

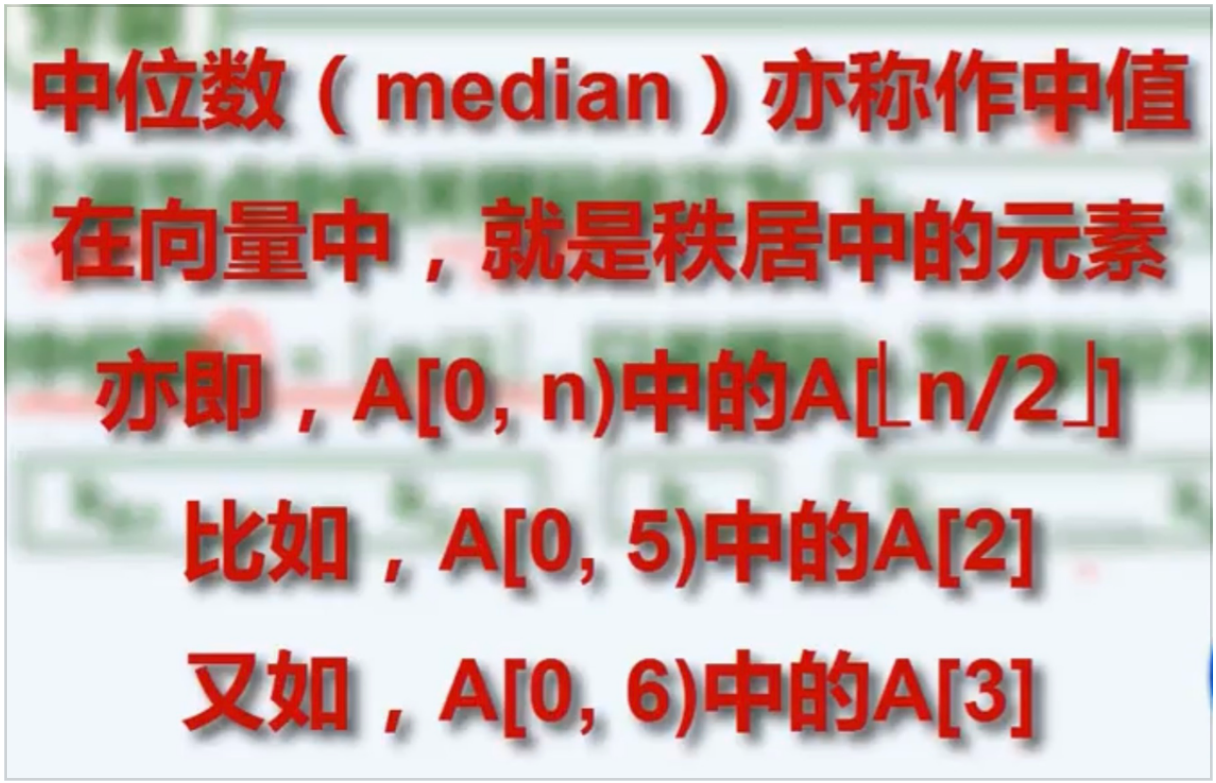
#数据结构邓神

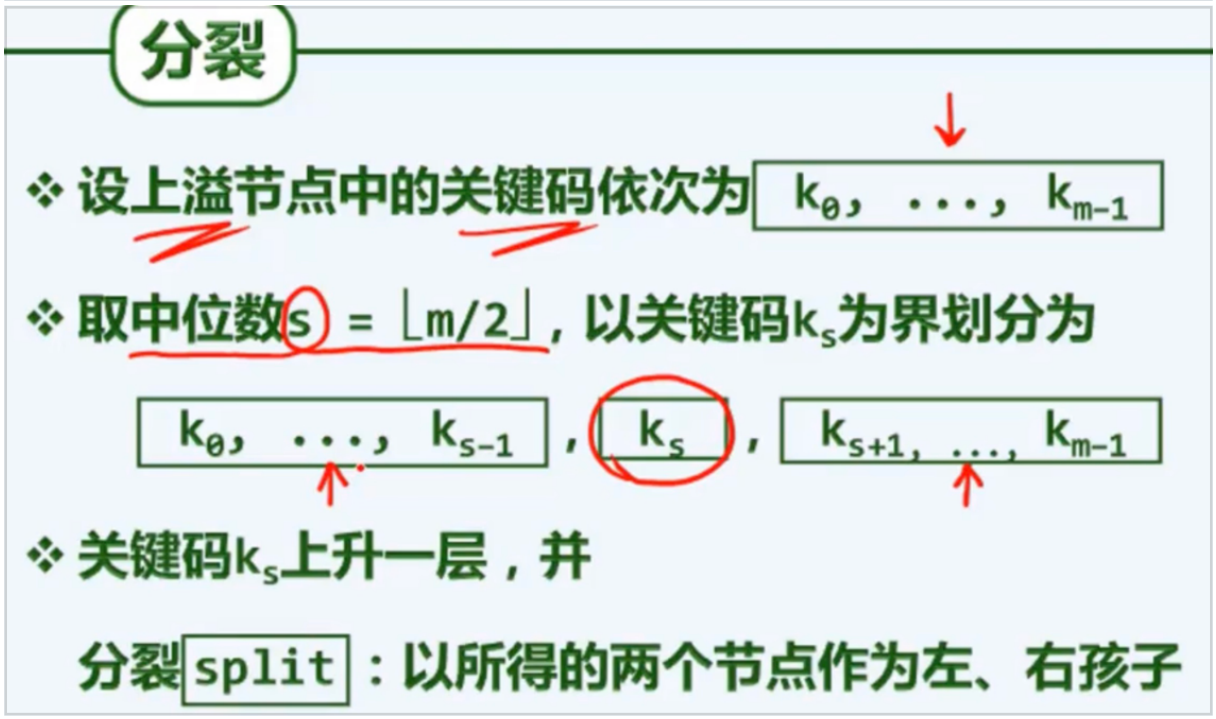
算法框架



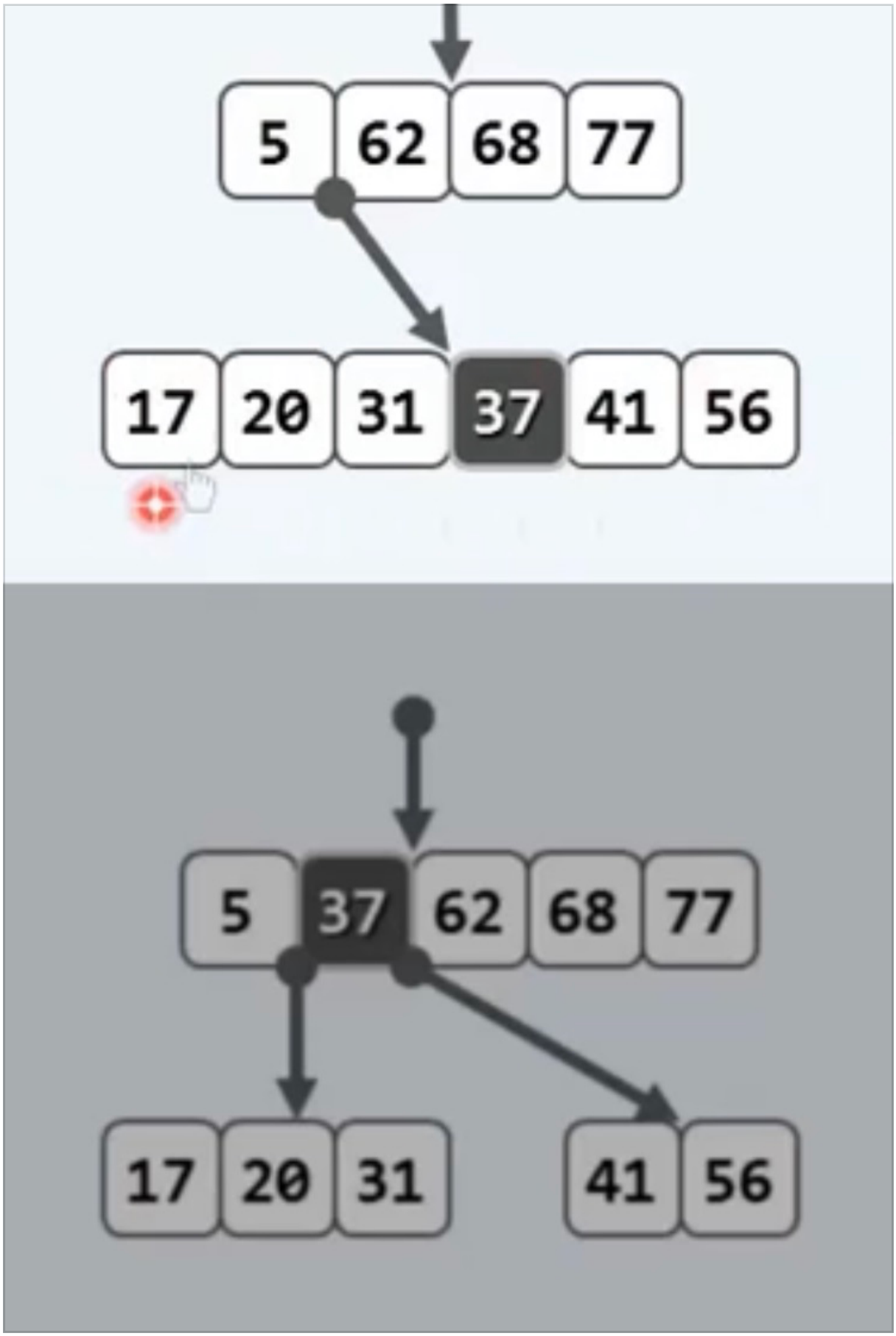

```
// insert
template <typename T> bool BTree<T>::insert(const T &e) {
    BTNodePosi(T) v = search(e);
    if (v) {
        return false;
    }
    Rank r = _hot->key.search(e);
    _hot->key.insert(r+1,e);
    _hot->child.insert(r+2,nullptr);
    _size++;
    solveOverFlow(_hot);
    return true;
}
```

分裂


中位数（median）亦称作中值
在向量中，就是秩居中的元素
亦即，A[0, n)中的A[⌊n/2⌋]
比如，A[0, 5)中的A[2]
又如，A[0, 6)中的A[3]


分裂
❖ 设上溢节点中的关键码依次为 $k_0, \ldots, k_{m-1}$
❖ 取中位数s = ⌊m/2⌋，以关键码k_s为界划分为
$k_0, \ldots, k_{s-1}$, k_s , $k_{s+1}, \ldots, k_{m-1}$
❖ 关键码k_s上升一层，并
分裂split：以所得的两个节点作为左、右孩子

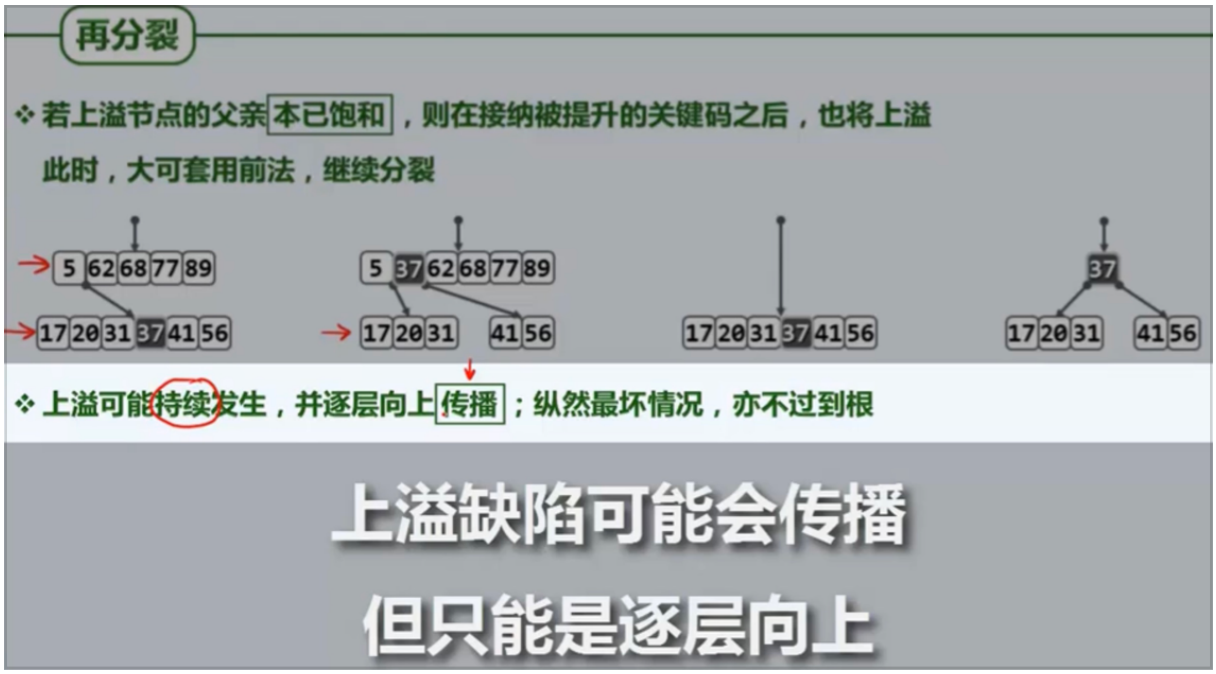
分裂实例：



再分裂

不难发现，这样的上溢可能会使父节点也发生溢出，同样的父节点，也需要执行一次分裂，那

么根节点怎么办呢?


再分裂
若上溢节点的父亲本已饱和，则在接纳被提升的关键码之后，也将上溢
此时，大可套用前法，继续分裂
5 62 68 77 89
17 20 31 37 41 56
5 37 62 68 77 89
17 20 31
41 56
17 20 31 37 41 56
37
17 20 31
41 56
上溢可能持续发生，并逐层向上传播；纵然最坏情况，亦不过到根
上溢缺陷可能会传播
但只能是逐层向上