06-B2

二叉搜索树

算法及实现：插入

邓俊辉

deng@tsinghua.edu.cn

算法

- 先借助search(e)

  确定插入位置及方向

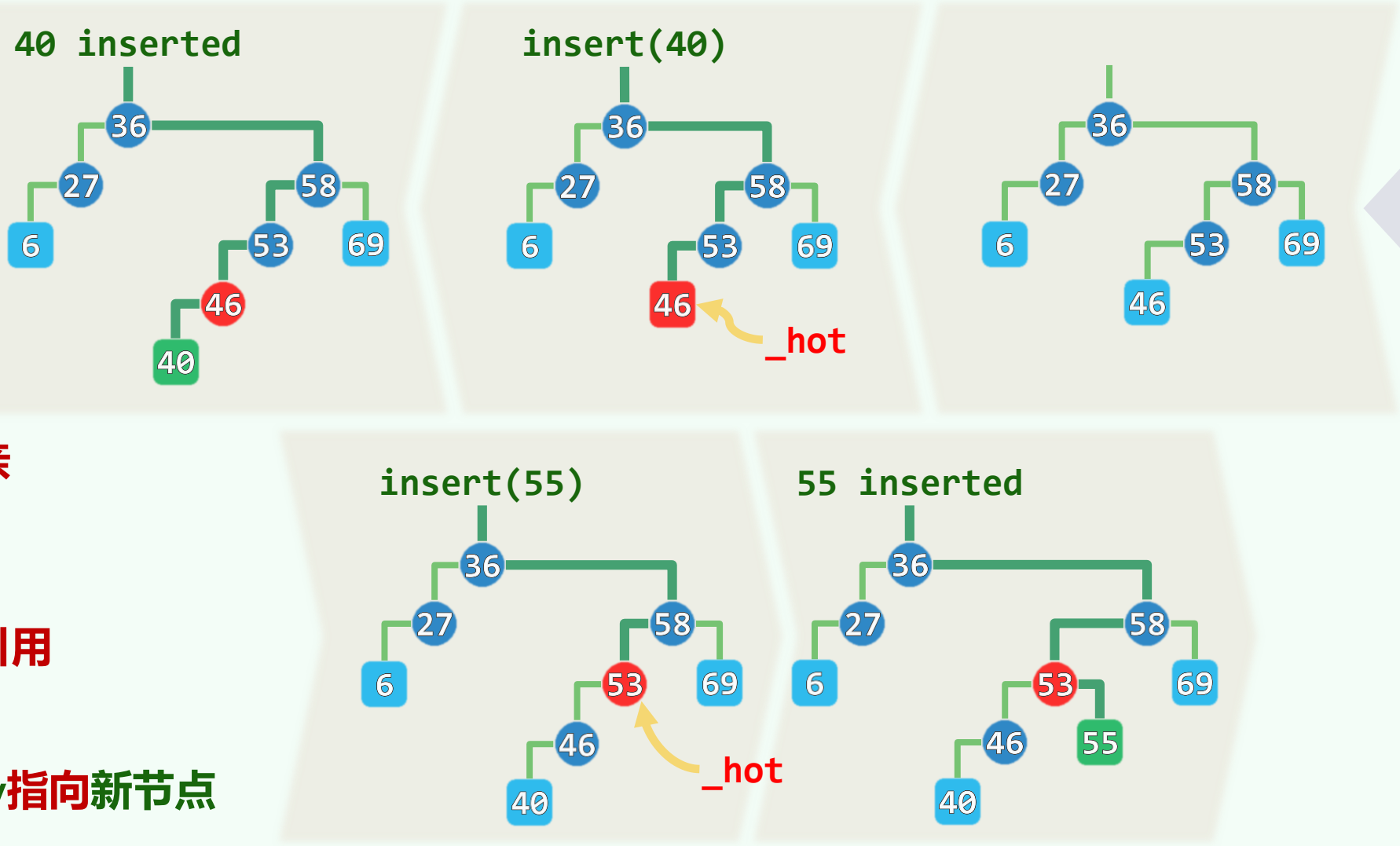
- 若e尚不存在，则再

  将新节点作为叶子插入

  - _hot为新节点的父亲
  - v = search(e)

    为_hot对新孩子的引用

- 于是，只需令_hot通过v指向新节点



实现

```
template <typename T> BinNodePosi<T> BST<T>::insert( const T & e ) {
   BinNodePosi<T> & x = search( e ); //通过查找
   if ( x ) return x; //确认目标不存在，并设置_hot
   ---------------------------------------------------
   x = new BinNode<T>( e, _hot ); //在x处创建新节点，以_hot为父亲
   _size++; x->updateHeightAbove(); //更新全树规模，以及历代祖先的高度
   return x; //新插入的节点，必为叶子
} //无论e是否存在于原树中，返回时总有x->data == e
```

❖ **时间主要消耗于search(e)和updateHeightAbove(x)；均线性正比于x的深度，不超过树高**