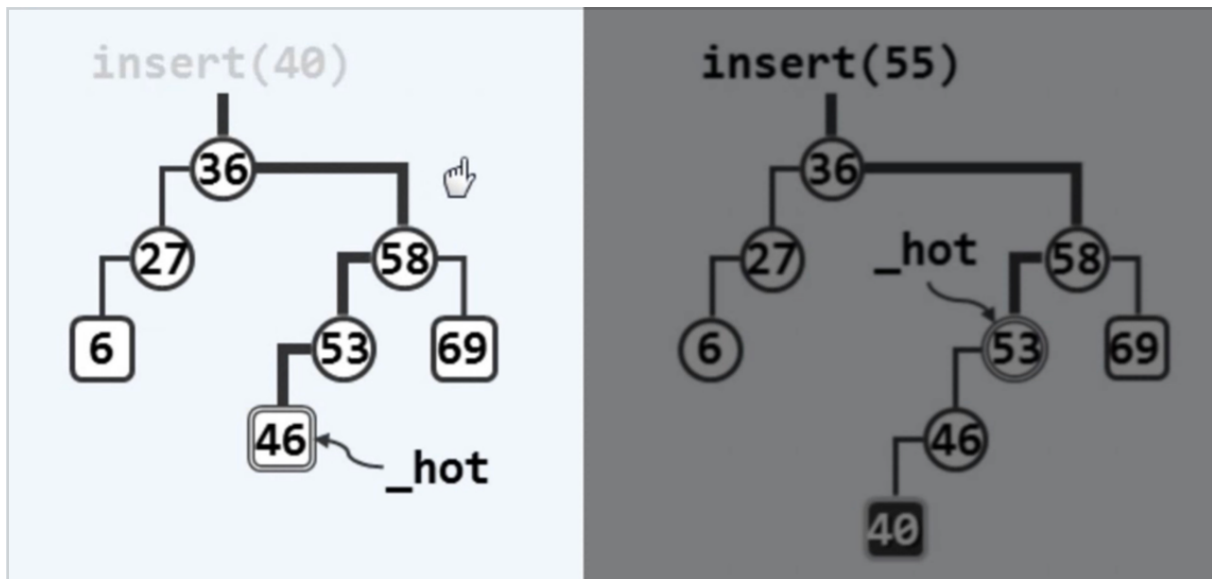


## 08B2 插入

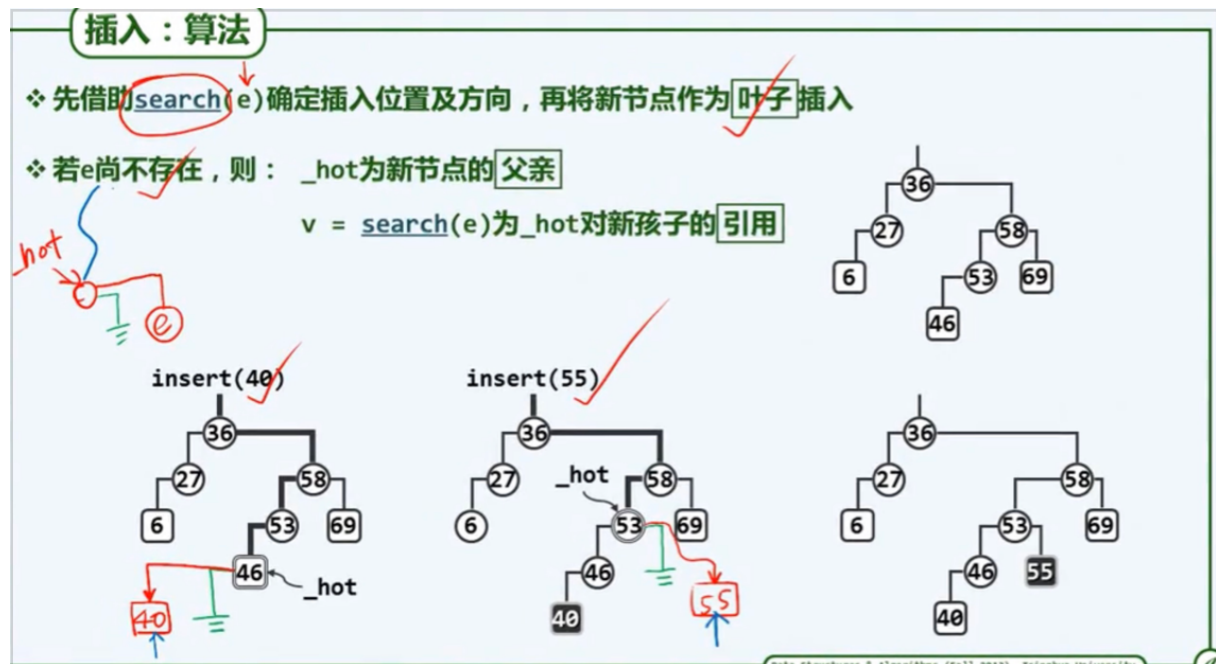
#数据结构邓神

### 算法



我们要插入40会失败于40

\_hot会指向46，我们只需要将40封装为一个新节点让46指向就可以



### 实现

## 插入：实现

```
❖ template <typename T> BinNodePosi(T) BST<T>::insert( const T & e ) {  
    BinNodePosi(T) & x = search( e ); //查找目标 (留意_hot的设置)  
    if ( !x ) { //既禁止雷同元素，故仅在查找失败时才实施插入操作  
        x = new BinNode<T>( e, _hot ); //在x处创建新节点，以_hot为父亲  
        _size++; updateHeightAbove( x ); //更新全树规模，更新x及其历代祖先的高度  
    }  
    return x; //无论e是否存在于原树中，至此总有x->data == e  
} //验证：对于首个节点插入之类的边界情况，均可正确处置
```

```
// insert  
template <typename T> BinNodePosi<T> BST<T>::insert(const T & e) {  
    BinNodePosi<T> & x = search(e);  
    if (!x){ // 禁止雷同!  
        x = new BinNode<T>(e,_hot);  
        _size++;  
        updateHeightAbove(x);  
    }  
    return x;  
}
```

时间复杂度：两个地方主要是 search updateHeightAbove

时间复杂度不超过  $O(h)$   $h$ 为高度