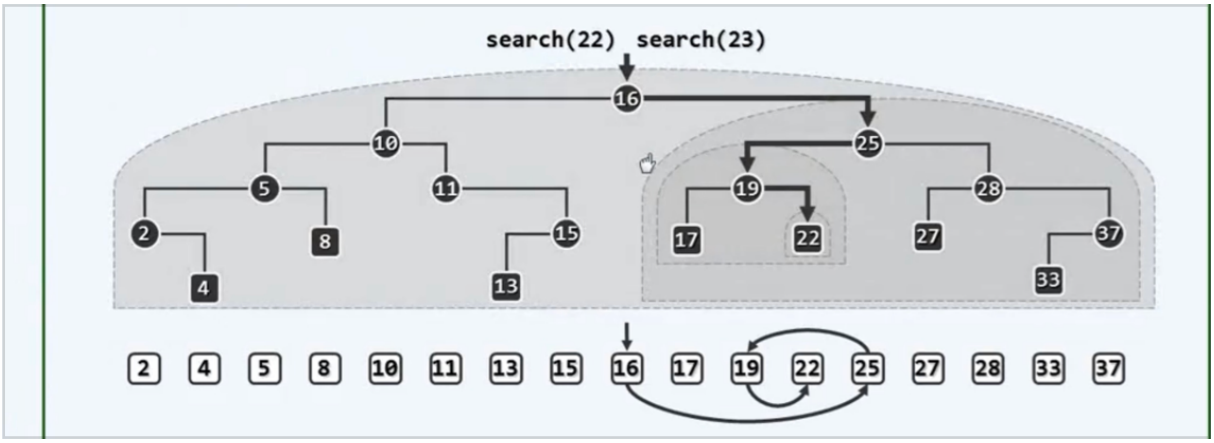
08B1 BST:查找

#数据结构邓神

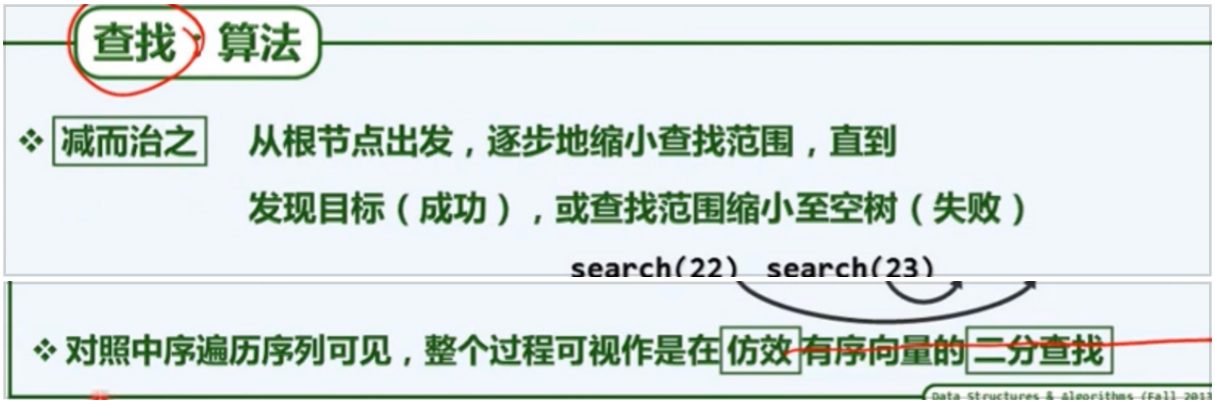
算法



理解

这就是树的二分查找?

好像跟向量的二分查找差不多



实现


```
// search
template <typename T> BinNodePosi<T> & BST<T>::search(const T & e){
    return searchIn(_root, e, _hot = nullptr);
}
// searchIn
template <typename T> static BinNodePosi<T> & searchIn(BinNodePosi<T> & v,
        const T & e,BinNodePosi<T> & hot){
    if (!v || (e == v->data)){
        return v;
    }
    hot = v;
    return searchIn((e < v->data)?v->lChild:v->rChild, e, hot);
}
```

语义｜hot 有什么用

查找：接口语义

❖ 返回的引用值： 成功时，指向一个关键码为e且真实存在的节点

失败时，指向最后一次试图转向的空节点NULL

❖ 失败时，不妨假想地将此空节点，转换为一个数值为e的哨兵节点

如此，依然满足BST的充要条件；而且更重要地...

❖ 无论成功与否：返回值总是等效地指向命中节点，而_hot总是指向命中节点的父亲