

本节内容

折半查找

王道考研/CSKAOYAN.COM

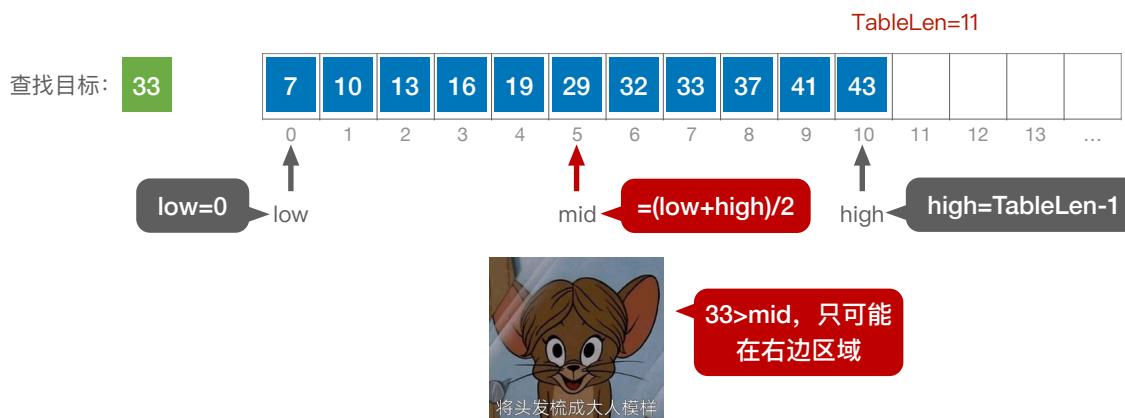
知识总览



王道考研/CSKAOYAN.COM

折半查找的算法思想

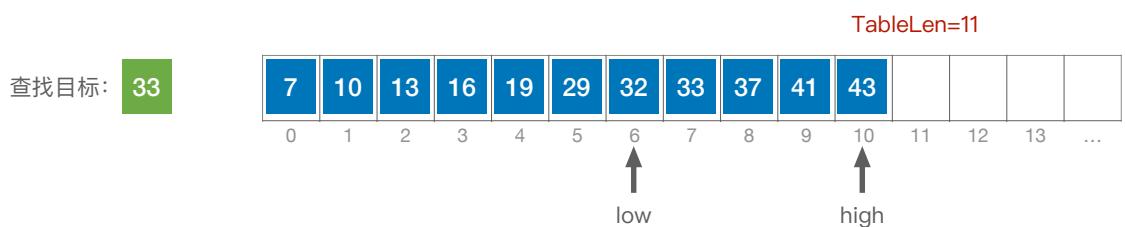
折半查找，又称“二分查找”，仅适用于**有序的顺序表**。



王道考研/CSKAOYAN.COM

折半查找的算法思想

折半查找，又称“二分查找”，仅适用于**有序的顺序表**。



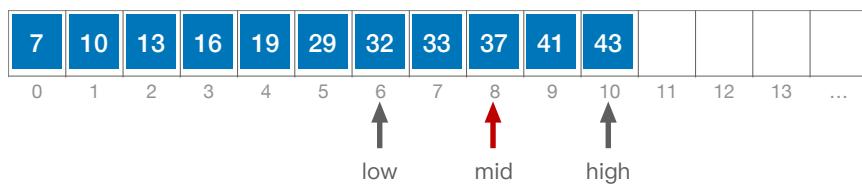
注：只有在 [low, high] 之间才有可能找到目标关键字

王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 33

TableLen=11



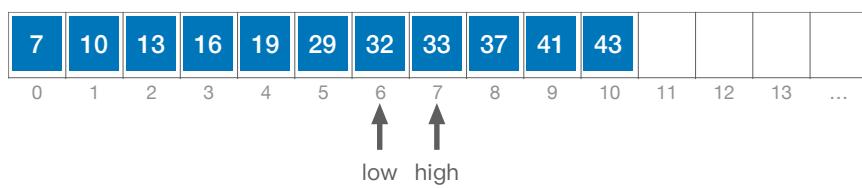
33<mid, 只可能
在左边区域

王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 33

TableLen=11

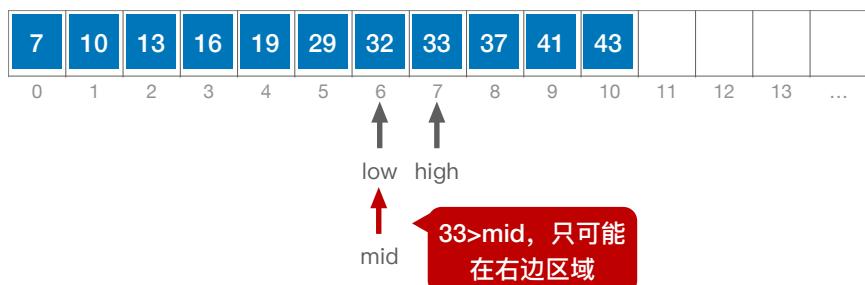


王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 33

TableLen=11

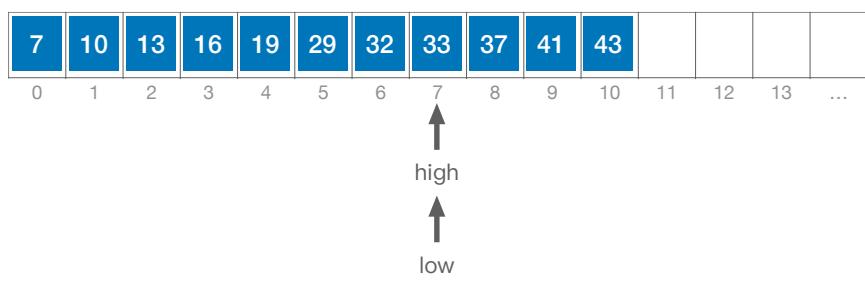


王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 33

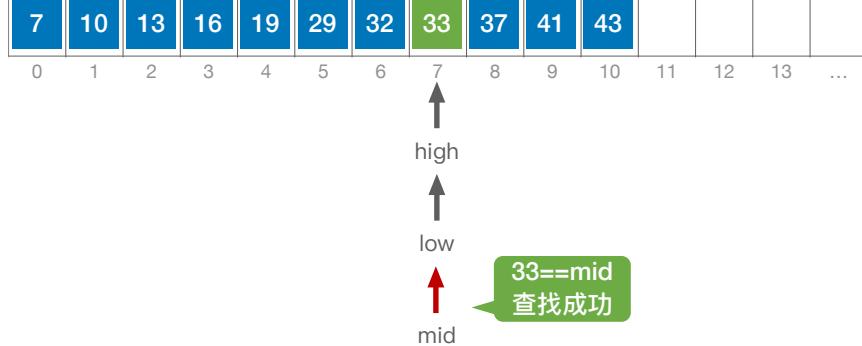
TableLen=11



王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 33



王道考研/CSKAOYAN.COM

折半查找的算法思想

查找目标： 12

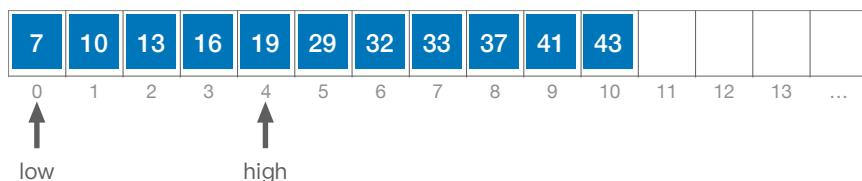


王道考研/CSKAOYAN.COM

折半查找的算法思想

TableLen=11

查找目标： 12

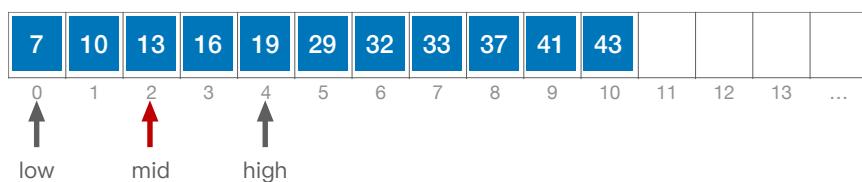


王道考研/CSKAOYAN.COM

折半查找的算法思想

TableLen=11

查找目标： 12

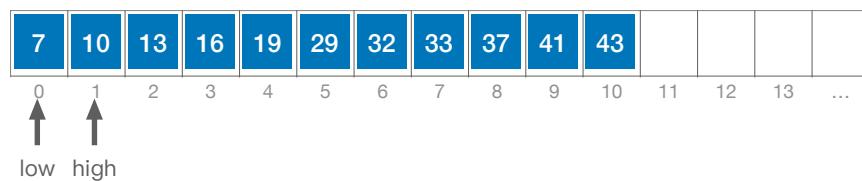


王道考研/CSKAOYAN.COM

折半查找的算法思想

TableLen=11

查找目标： 12

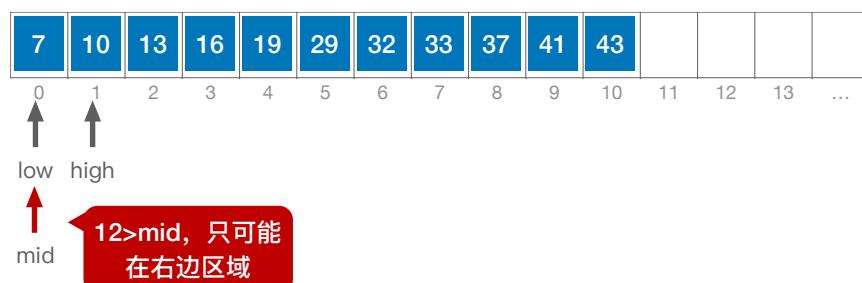


王道考研/CSKAOYAN.COM

折半查找的算法思想

TableLen=11

查找目标： 12



王道考研/CSKAOYAN.COM

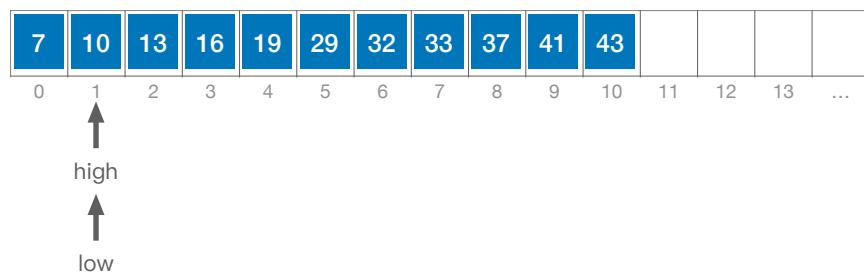
折半查找的算法思想



折半查找，又称“二分查找”，仅适用于**有序的顺序表**。

查找目标： 12

TableLen=11



王道考研/CSKAOYAN.COM

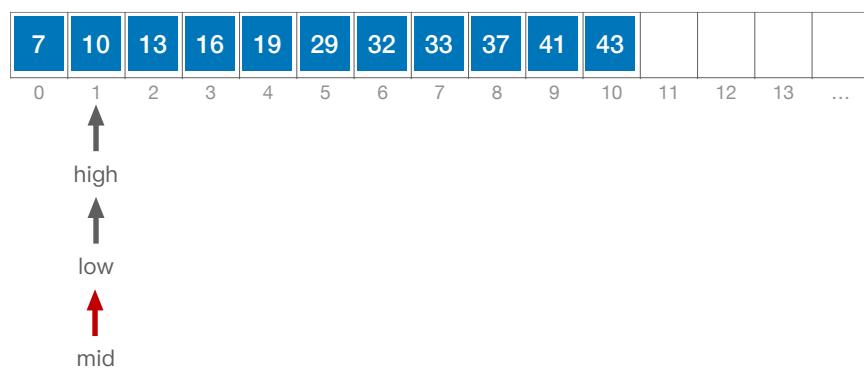
折半查找的算法思想



折半查找，又称“二分查找”，仅适用于**有序的顺序表**。

查找目标： 12

TableLen=11



王道考研/CSKAOYAN.COM

折半查找的算法思想

折半查找，又称“二分查找”，仅适用于**有序的顺序表**。



王道考研/CSKAOYAN.COM

折半查找的实现

```

typedef struct{           //查找表的数据结构 (顺序表)
    ELEMTYPE *elem;        //动态数组基址
    int TableLen;          //表的长度
}SSTable;

折半查找，又称“二分查找”，
仅适用于有序的顺序表。

```

顺序表拥有随机访问的特性，链表没有

```

//折半查找
int Binary_Search(SSTable L, ELEMTYPE key){
    int low=0, high=L.TableLen-1, mid;
    while(low<=high){
        mid=(low+high)/2;           //取中间位置
        if(L.elem[mid]==key)       //查找成功则返回所在位置
            return mid;
        else if(L.elem[mid]>key)   //从前半部分继续查找
            high=mid-1;
        else                         //从后半部分继续查找
            low=mid+1;
    }
    return -1;                   //查找失败，返回-1
}

```

TableLen=11

查找目标： 33



王道考研/CSKAOYAN.COM

查找效率分析



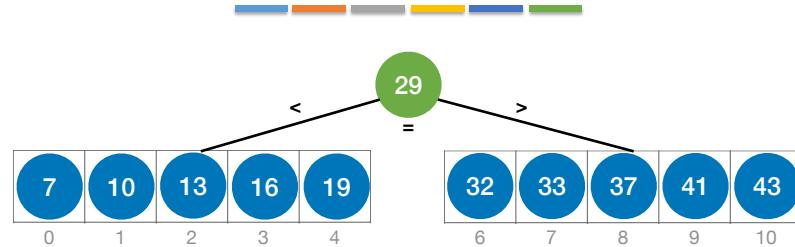
王道考研/CSKAOYAN.COM

查找效率分析



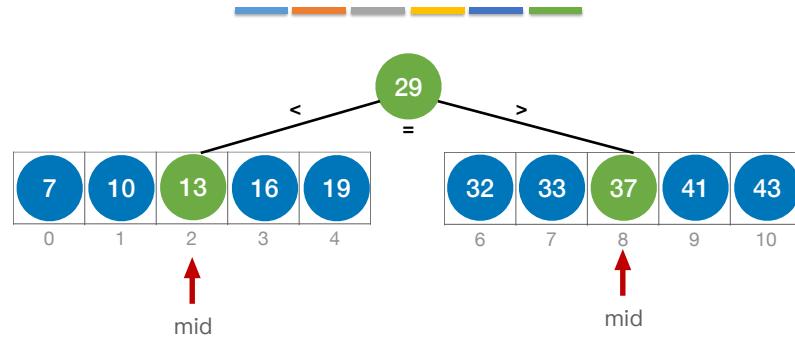
王道考研/CSKAOYAN.COM

查找效率分析



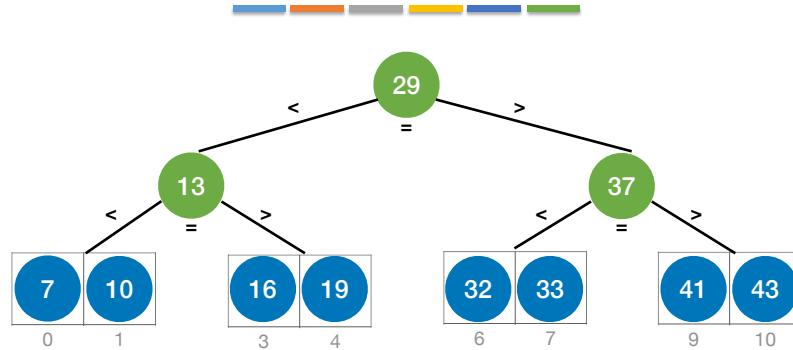
王道考研/CSKAOYAN.COM

查找效率分析



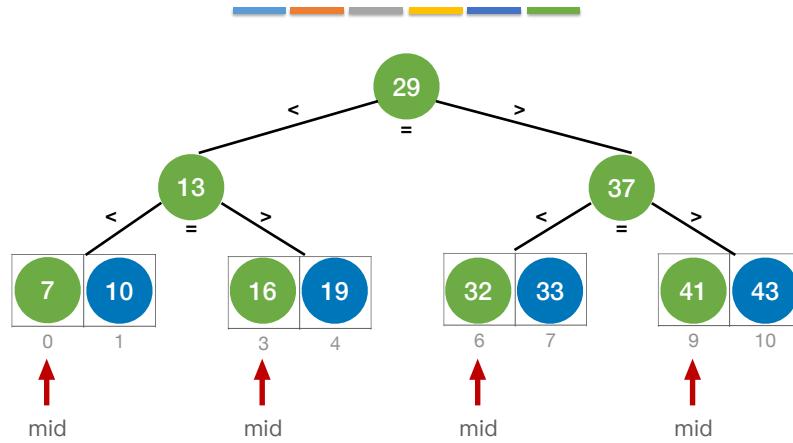
王道考研/CSKAOYAN.COM

查找效率分析



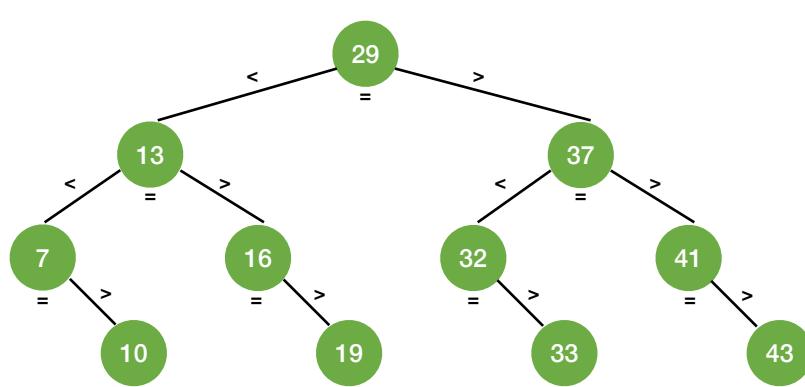
王道考研/CSKAOYAN.COM

查找效率分析



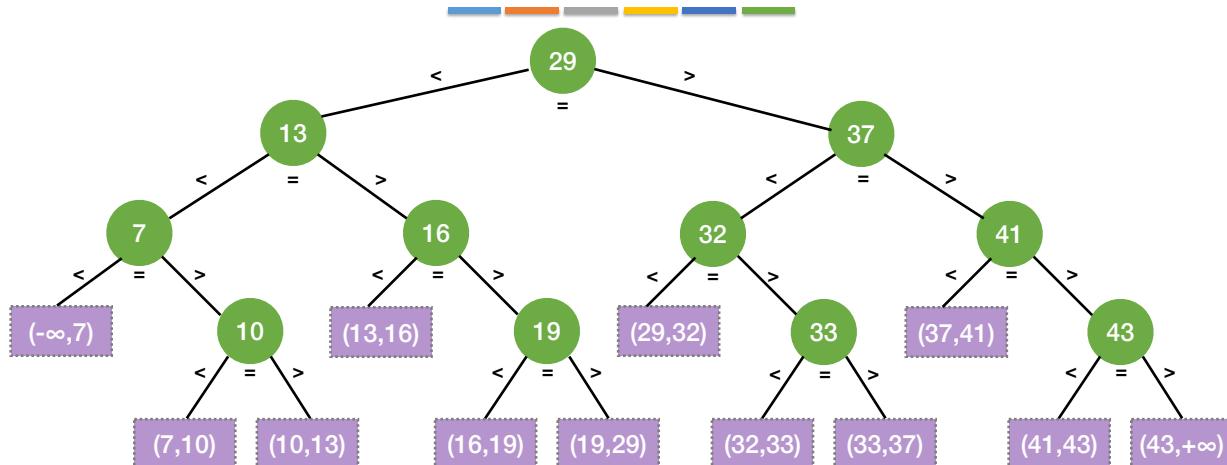
王道考研/CSKAOYAN.COM

查找效率分析



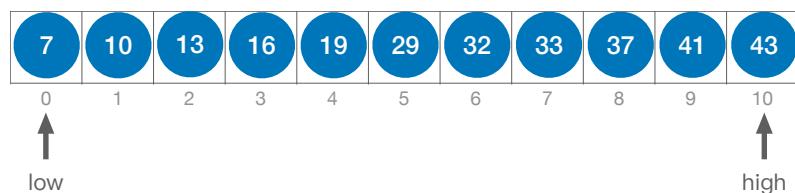
王道考研/CSKAOYAN.COM

查找效率分析



王道考研/CSKAOYAN.COM

折半查找判定树的构造



王道考研/CSKAOYAN.COM

折半查找判定树的构造

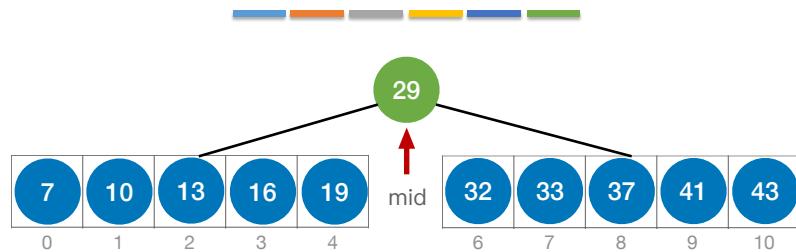


$$mid = \lfloor (low + high)/2 \rfloor$$

如果当前low和high之间有奇数个元素，则 mid 分隔后，左右两部分元素个数相等

王道考研/CSKAOYAN.COM

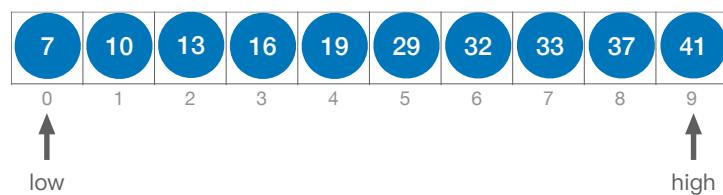
折半查找判定树的构造



如果当前low和high之间有奇数个元素，则 mid 分隔后，左右两部分元素个数相等

王道考研/CSKAOYAN.COM

折半查找判定树的构造



王道考研/CSKAOYAN.COM

折半查找判定树的构造

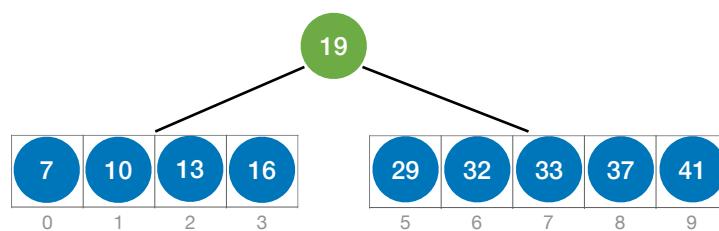


$$mid = \lfloor (low + high)/2 \rfloor$$

如果当前low和high之间有偶数个元素，则 mid 分隔后，左半部分比右半部分少一个元素

王道考研/CSKAOYAN.COM

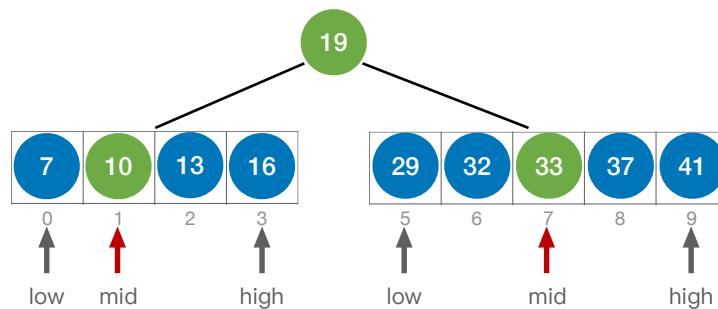
折半查找判定树的构造



如果当前low和high之间有偶数个元素，则 mid 分隔后，左半部分比右半部分少一个元素

王道考研/CSKAOYAN.COM

折半查找判定树的构造

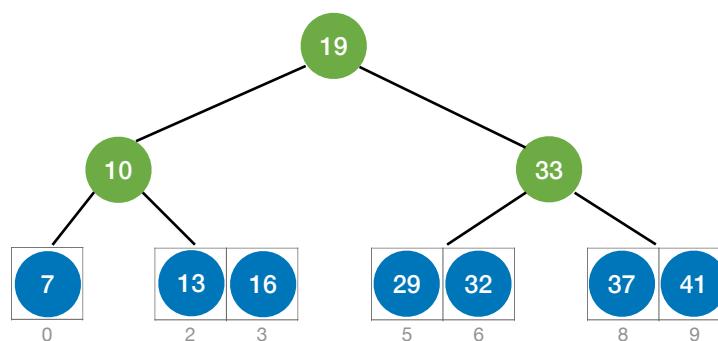


$$mid = \lfloor (low + high)/2 \rfloor$$

如果当前low和high之间有奇数个元素，则mid分隔后，左右两部分元素个数相等
如果当前low和high之间有偶数个元素，则mid分隔后，左半部分比右半部分少一个元素

王道考研/CSKAOYAN.COM

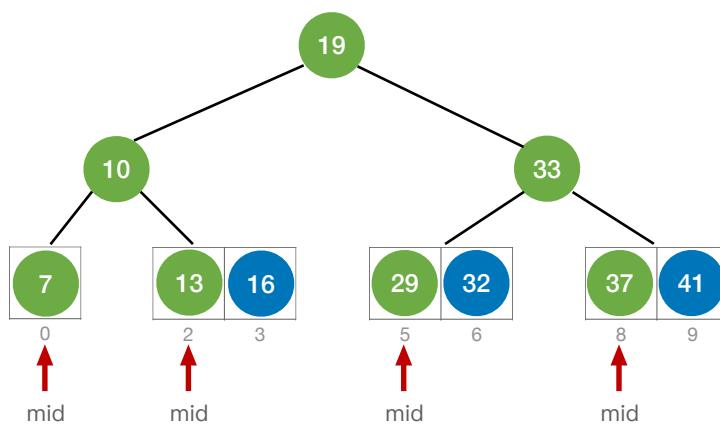
折半查找判定树的构造



如果当前low和high之间有奇数个元素，则mid分隔后，左右两部分元素个数相等
如果当前low和high之间有偶数个元素，则mid分隔后，左半部分比右半部分少一个元素

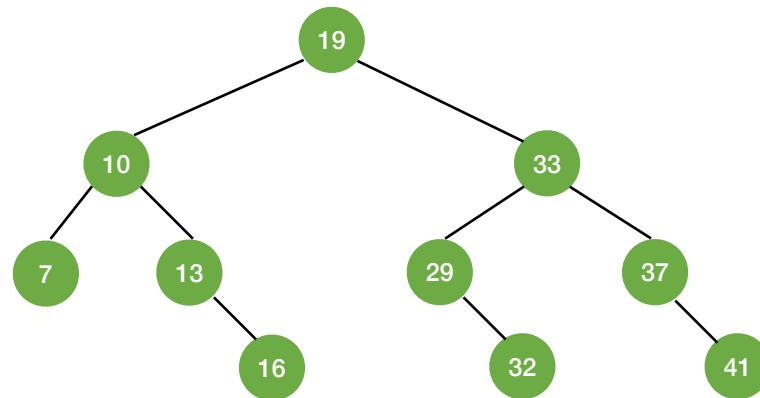
王道考研/CSKAOYAN.COM

折半查找判定树的构造



王道考研/CSKAOYAN.COM

折半查找判定树的构造



如果当前low和high之间有奇数个元素，则 mid 分隔后，左右两部分元素个数相等
如果当前low和high之间有偶数个元素，则 mid 分隔后，左半部分比右半部分少一个元素

王道考研/CSKAOYAN.COM

折半查找判定树的构造

如果当前low和high之间有奇数个元素，则 mid 分隔后，**左右两部分元素个数相等**
如果当前low和high之间有偶数个元素，则 mid 分隔后，**左半部分比右半部分少一个元素**

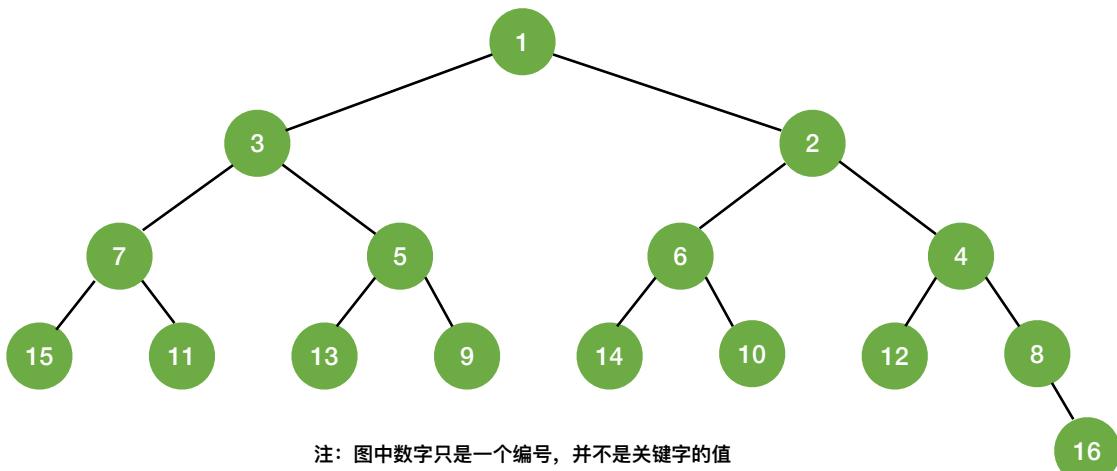


折半查找的判定树中，若 $mid = \lfloor (low + high)/2 \rfloor$ ，则对于任何一个结点，必有：
右子树结点数 - 左子树结点数 = 0 或 1

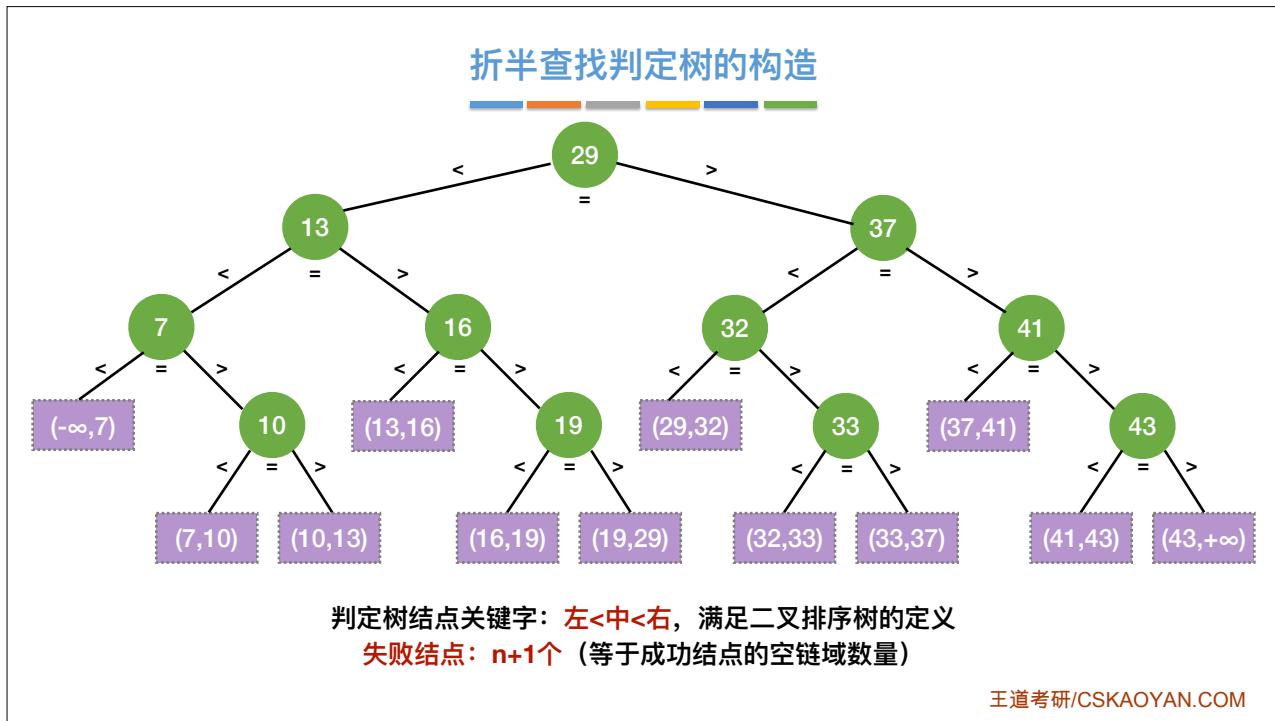
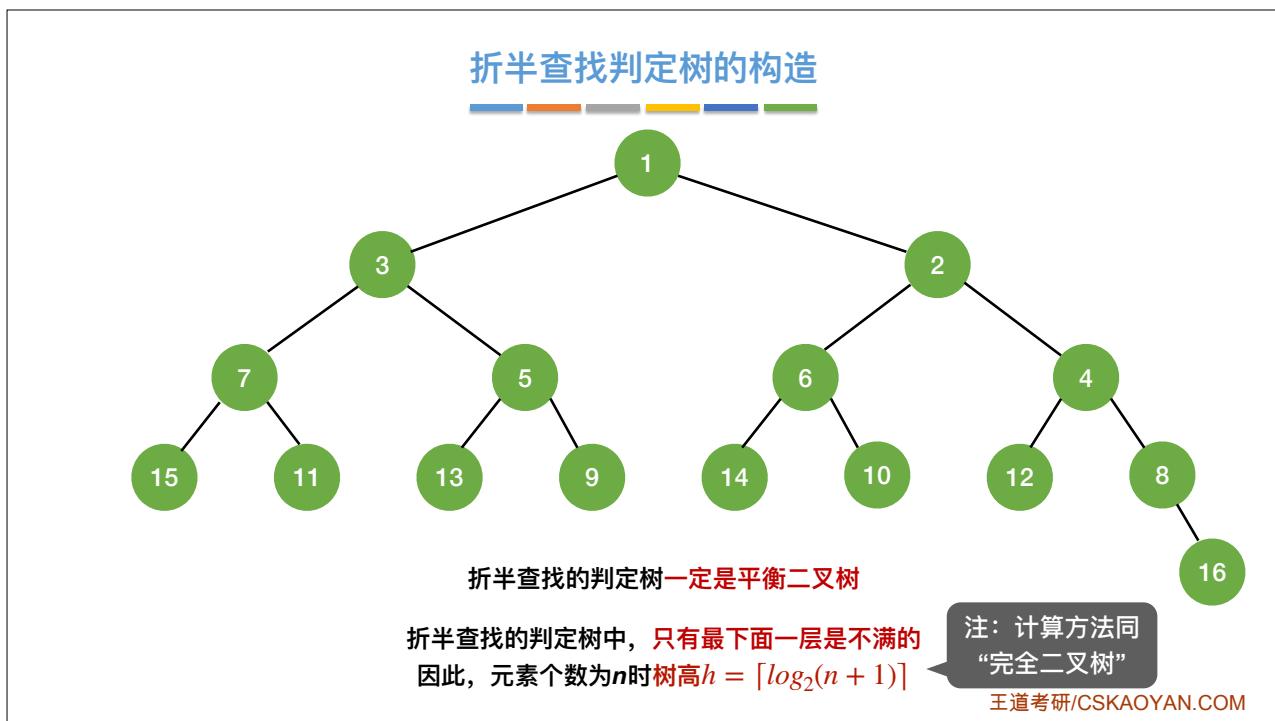
王道考研/CSKAOYAN.COM

折半查找判定树的构造

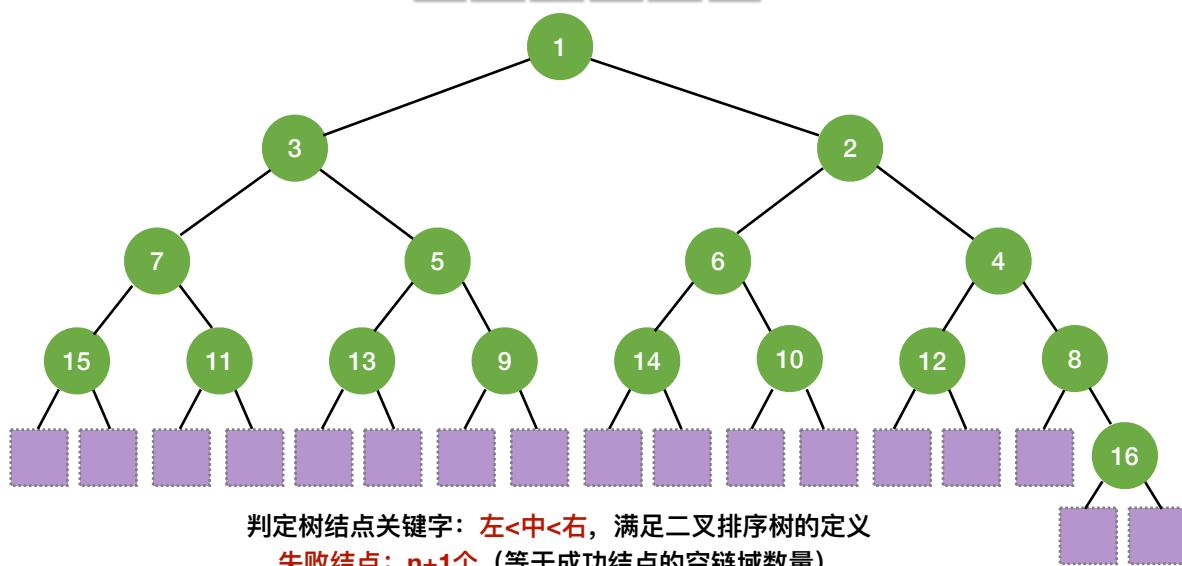
练习：若 $mid = \lfloor (low + high)/2 \rfloor$ ，画出含1个元素、2个元素、3个元素....16个元素 的查找表对应的折半查找判定树，注：暂不考虑失败结点 (Key: 右子树结点数 - 左子树结点数 = 0 或 1)



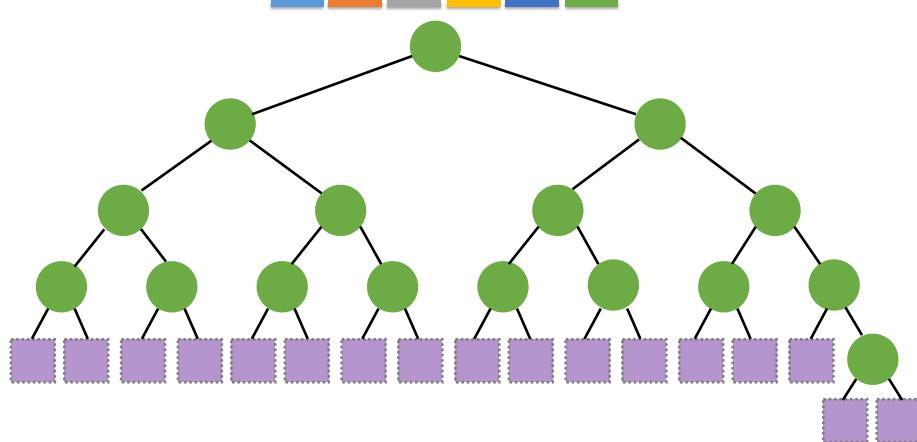
王道考研/CSKAOYAN.COM

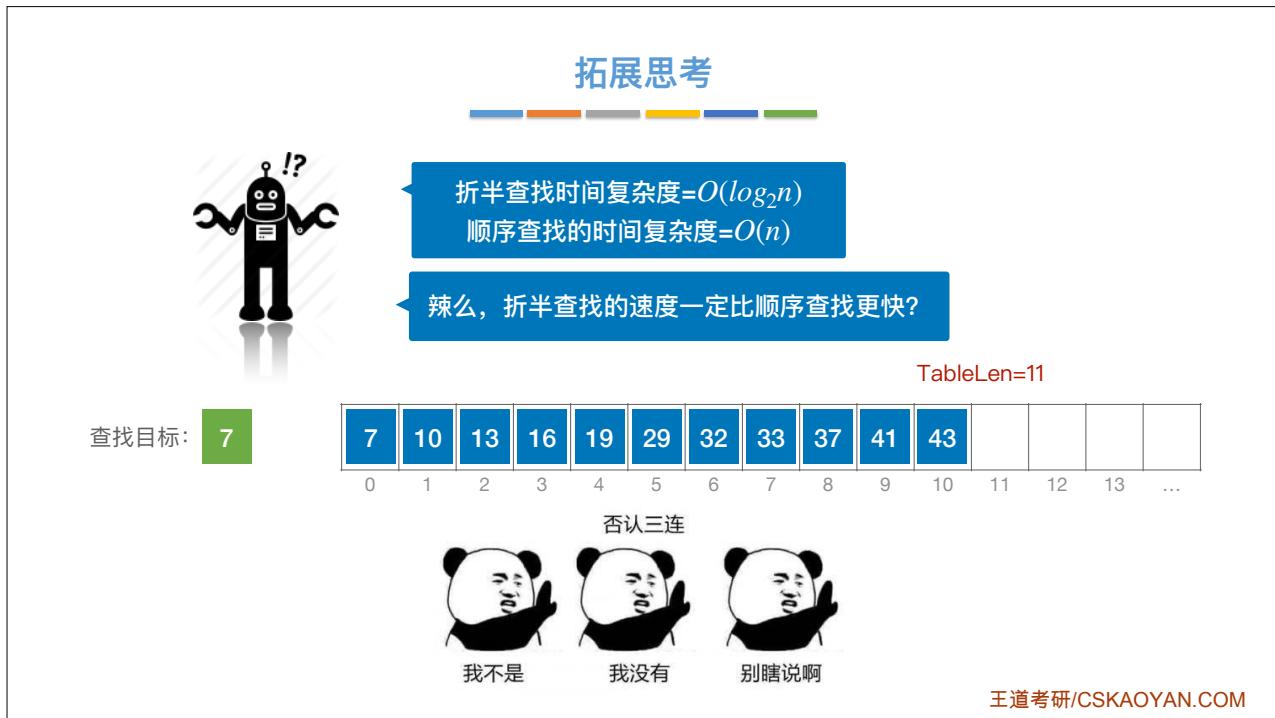
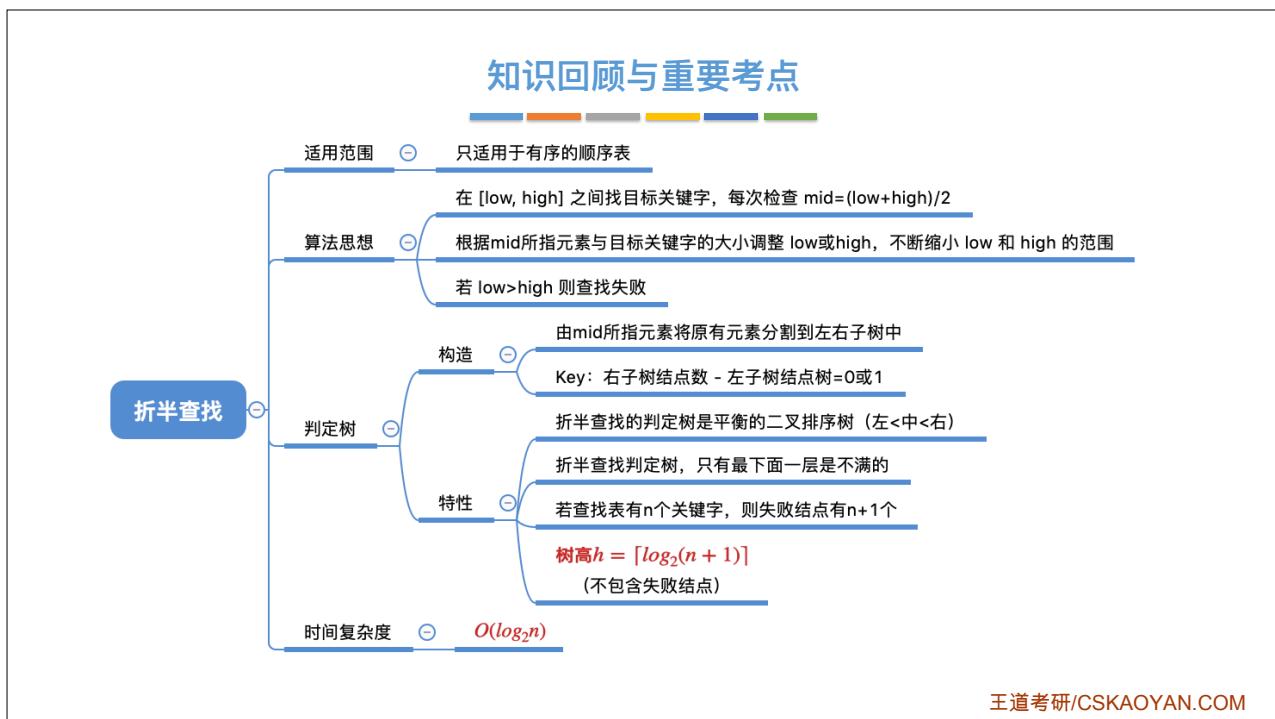


折半查找判定树的构造



折半查找的查找效率





拓展思考



如果 $mid = \lceil (low + high)/2 \rceil$
辣么，判定树是什么亚子？



$$mid = \lceil (low + high)/2 \rceil$$

如果当前low和high之间有奇数个元素，则 mid 分隔后，左右两部分元素个数相等

王道考研/CSKAOYAN.COM

拓展思考



如果 $mid = \lceil (low + high)/2 \rceil$
辣么，判定树是什么亚子？



$$mid = \lceil (low + high)/2 \rceil$$

如果当前low和high之间有偶数个元素，则 mid 分隔后，左半部分比右半部分多一个元素

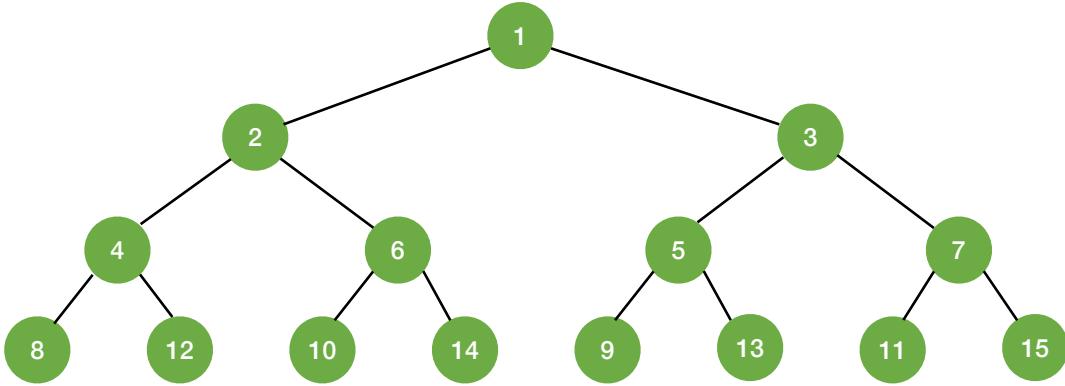
王道考研/CSKAOYAN.COM

拓展思考



折半查找的判定树中，若 $mid = \lceil (low + high)/2 \rceil$ ，则对于任何一个结点，必有：

左子树结点数-右子树结点数=0或1



注：图中数字只是一个编号，并不是关键字的值

王道考研/CSKAOYAN.COM