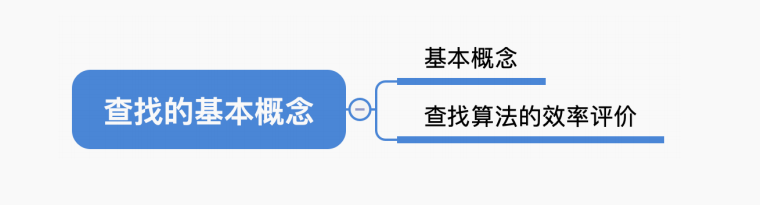
查找基本概念



查找的基本概念：

**查找**——在数据集合中**寻找满⾜某种条件的数据元素**的过程称为查找

**查找表**（查找结构）——⽤于**查找的数据集合**称为查找表，它由**同⼀类型的数据元素**（或记录）组成

**关键字**——数据元素中**唯⼀标识**该元素的某个数据项的值，使⽤**基于关键字的查找**，查找结果应该是**唯⼀**的。

对查找表的常⻅操作：

①查找符合条件的数据元素

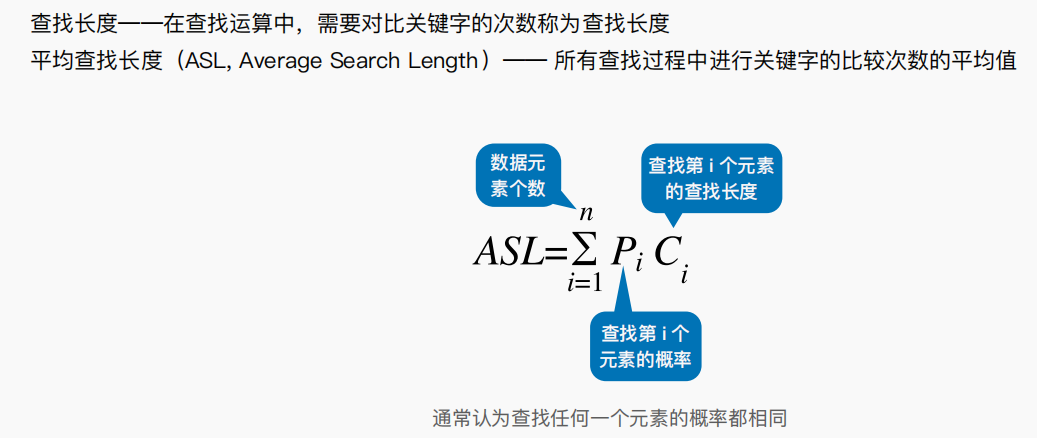
②插⼊、删除某个数据元素



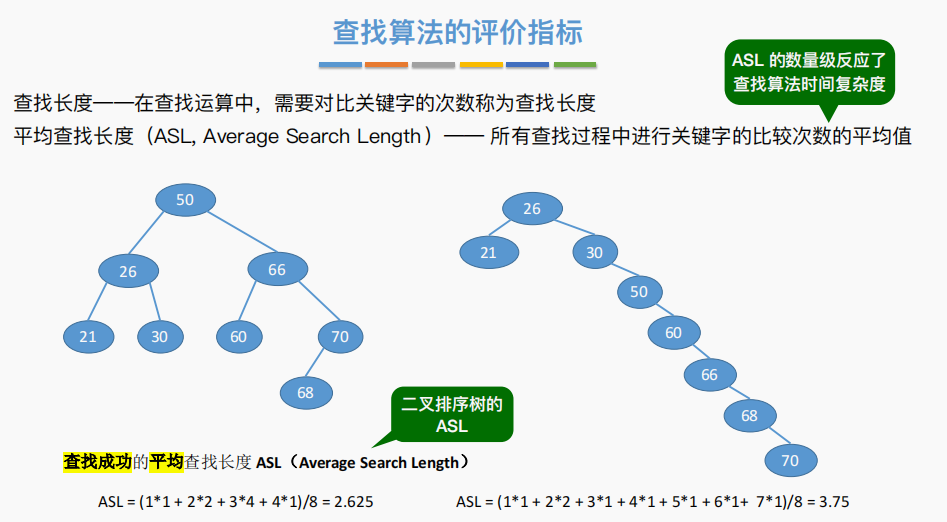
查找算法的评价指标：

**查找⻓度**——在查找运算中，需要**对⽐关键字的次数**称为查找⻓度

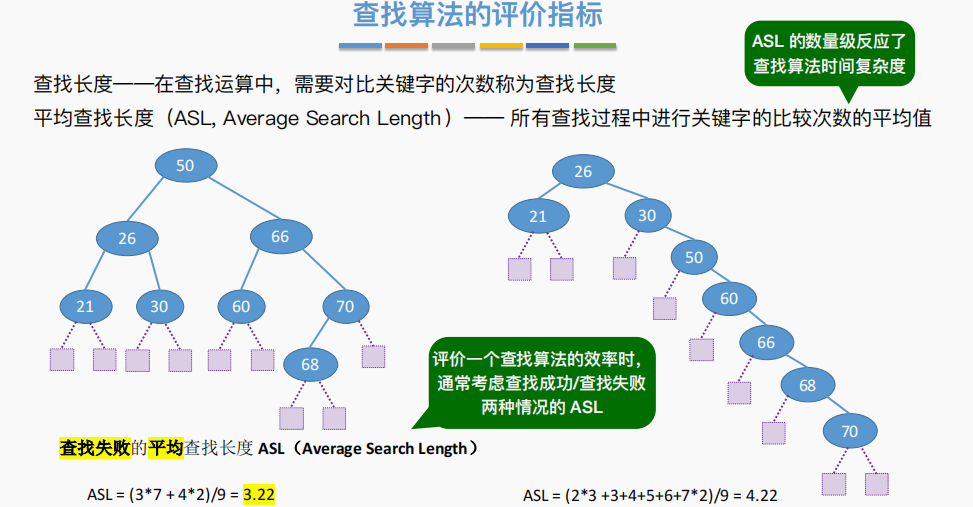
**平均查找⻓度**（ASL, Average Search Length）—— 所有查找过程中进⾏关键字的**⽐较次数的平均值**

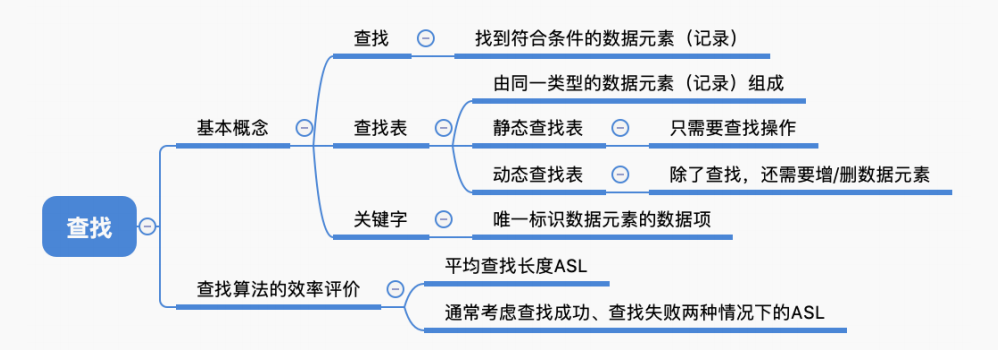


查找成功的平均查找长度ASL=1/n\*级数求和（所在层数\*个数）

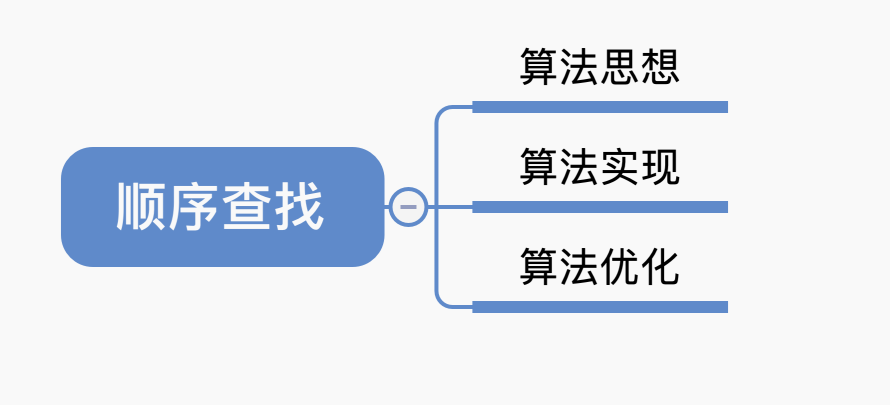


查找成功的平均查找长度ASL=1/n\*级数求和（（所在层数-1）\*失败结点个数）





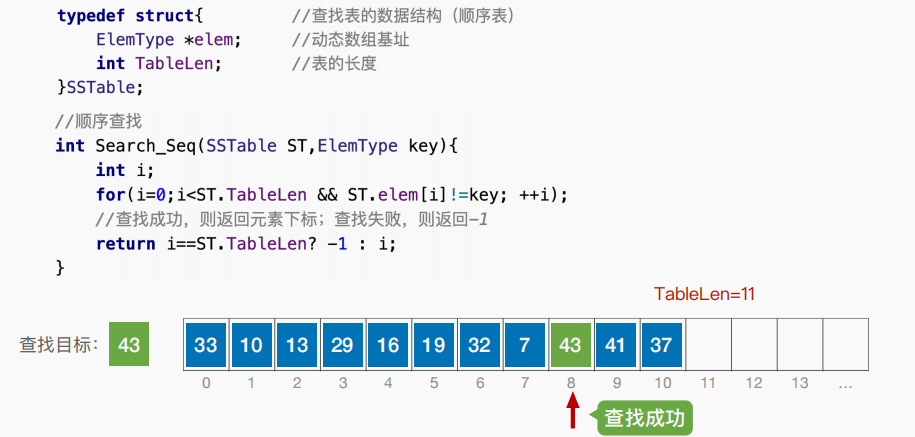
顺序查找



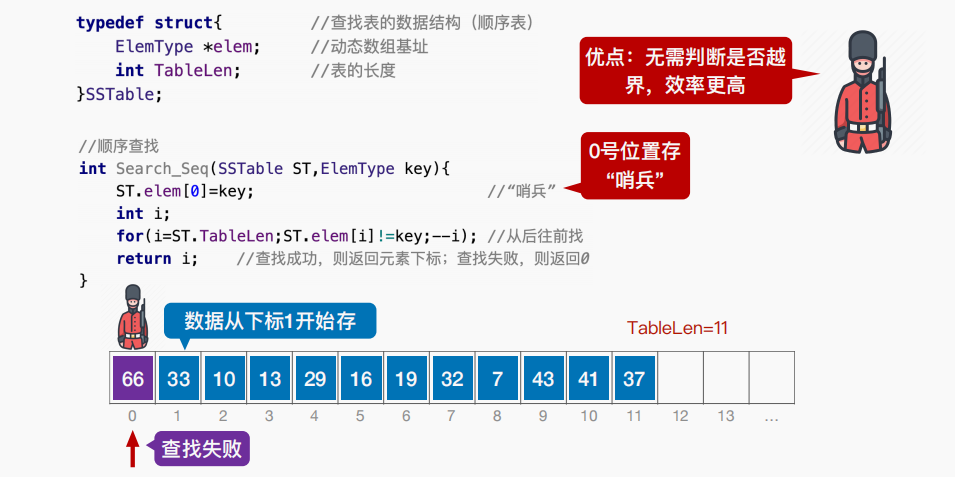
顺序查找，⼜叫“**线性查找**”，通常⽤于**线性表**。

算法思想：从头到尾挨个找（或者反过来）

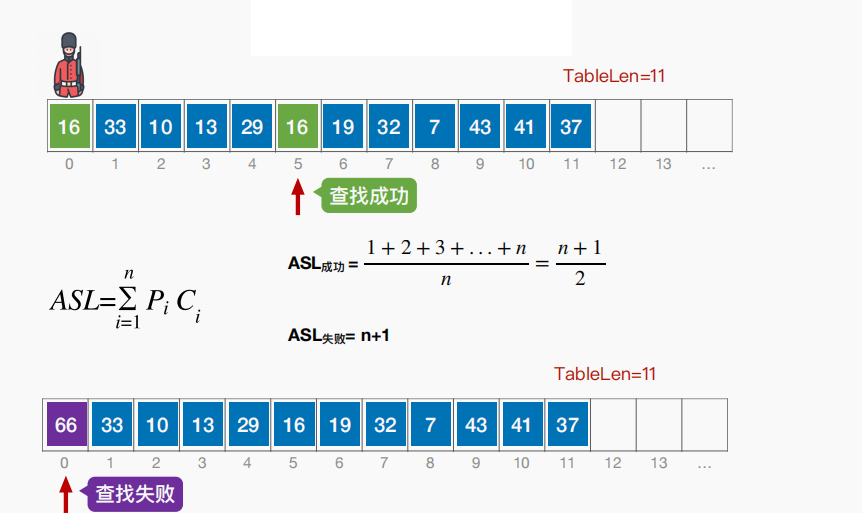
顺序查找的实现：



顺序查找的实现（哨兵）：

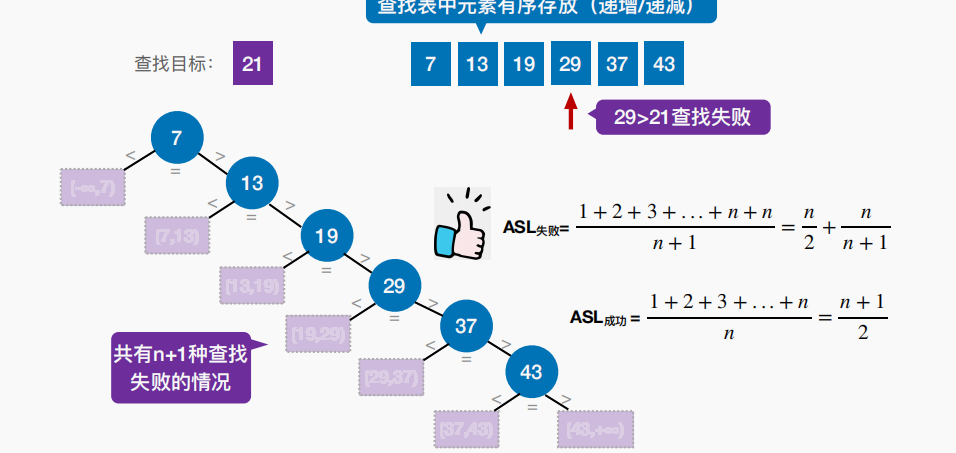


查询效率分析：

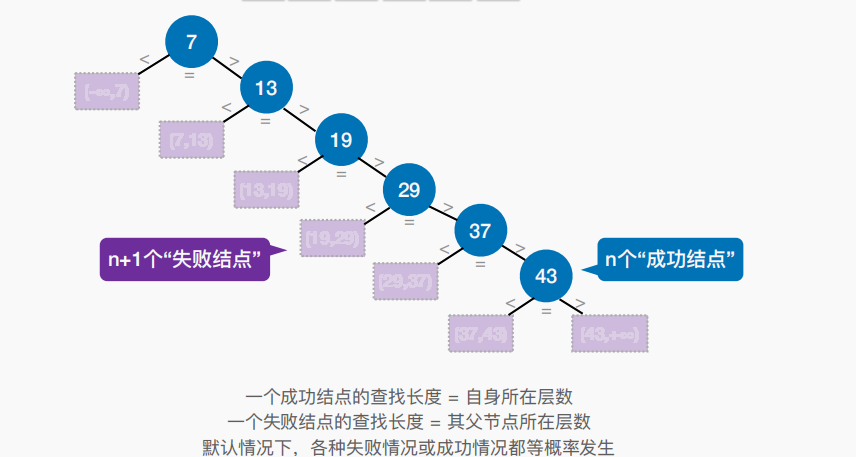


顺序查找的优化（对有序表）：





⽤查找判定树分析ASL:

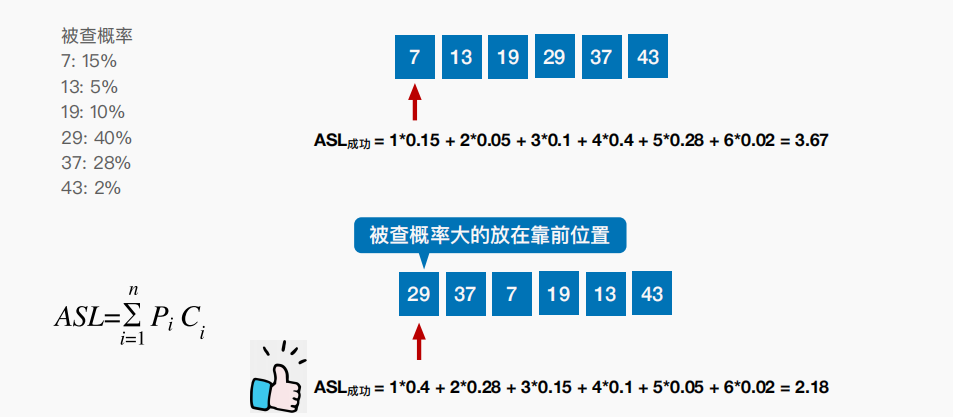


⼀个**成功结点**的查找⻓度 = **⾃身所在层数**

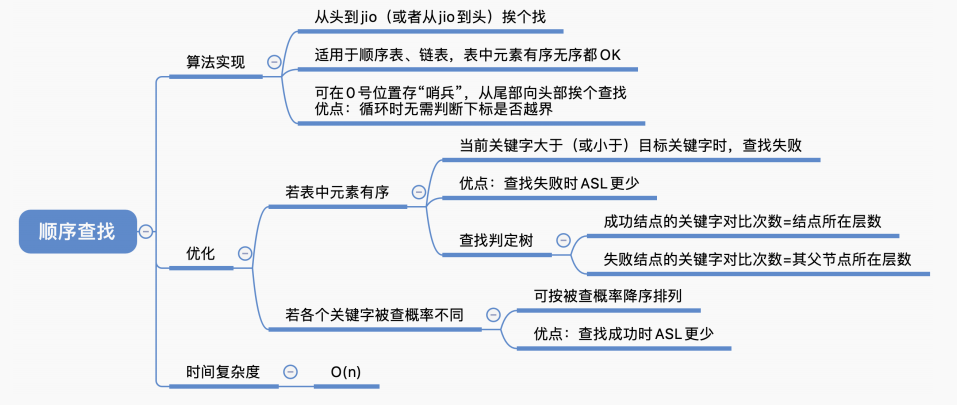
⼀个**失败结点**的查找⻓度 = **其⽗节点所在层数**

默认情况下，各种失败情况或成功情况都**等概率发⽣**

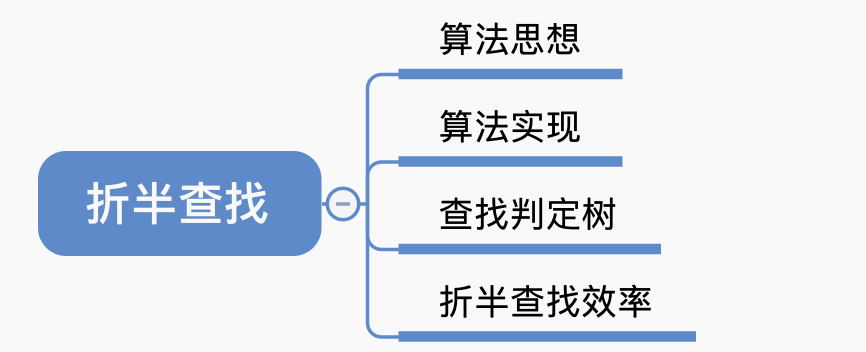
顺序查找的优化（被查概率不相等）：



总结：

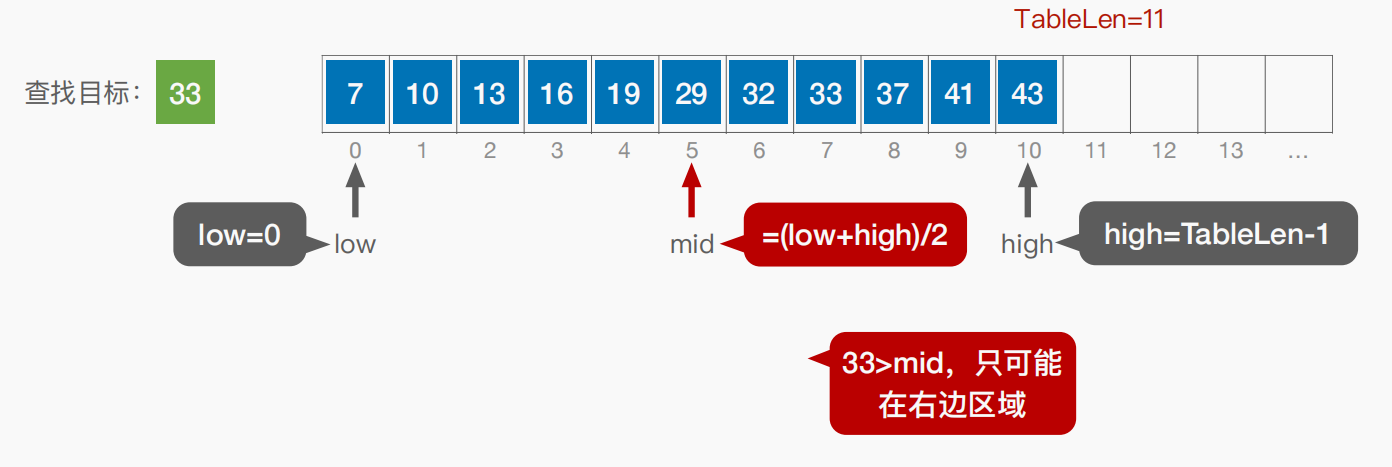


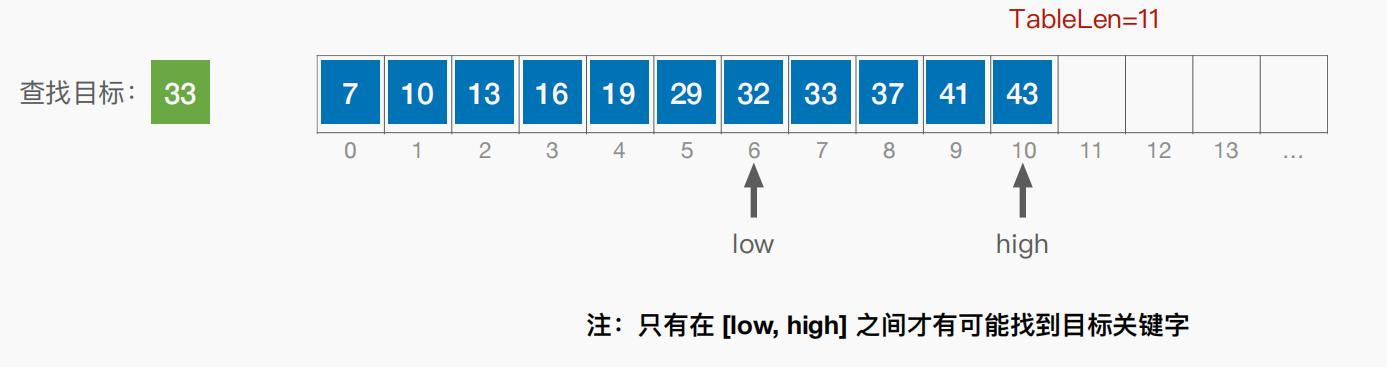
折半查找

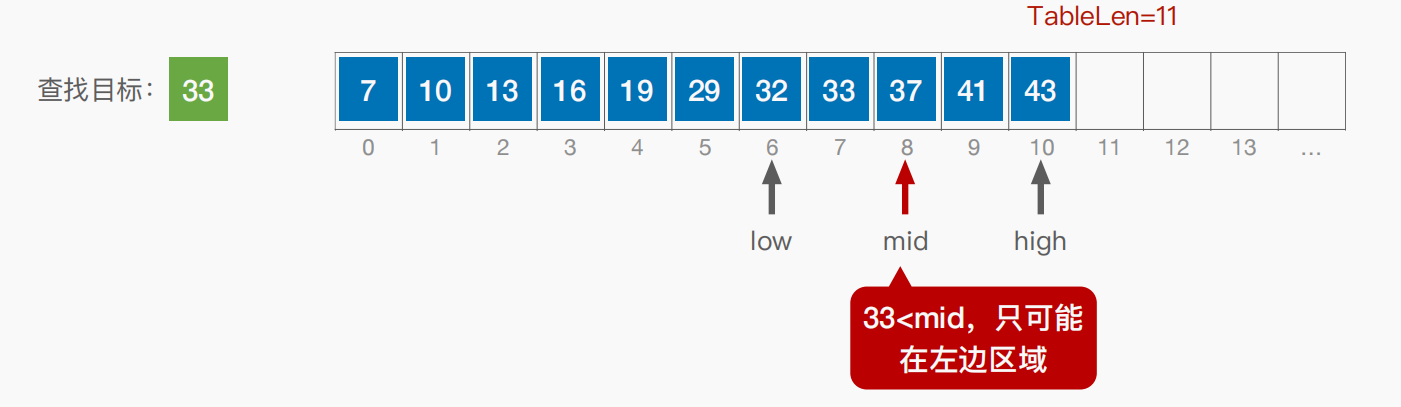


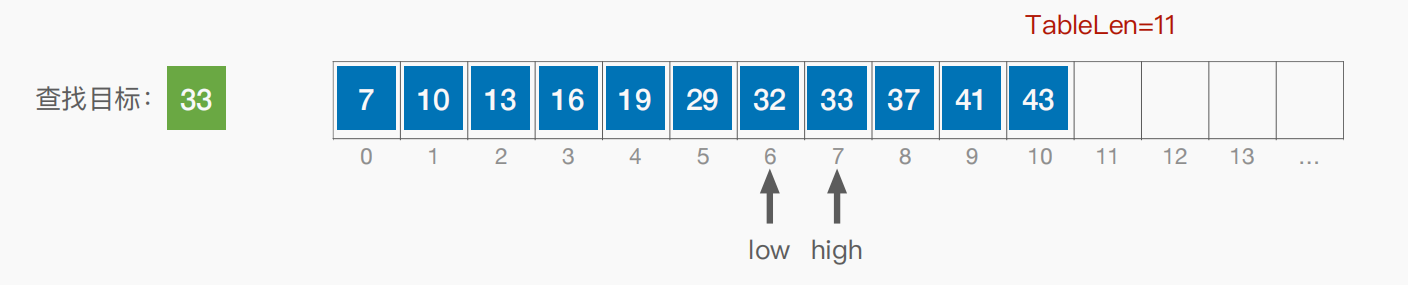
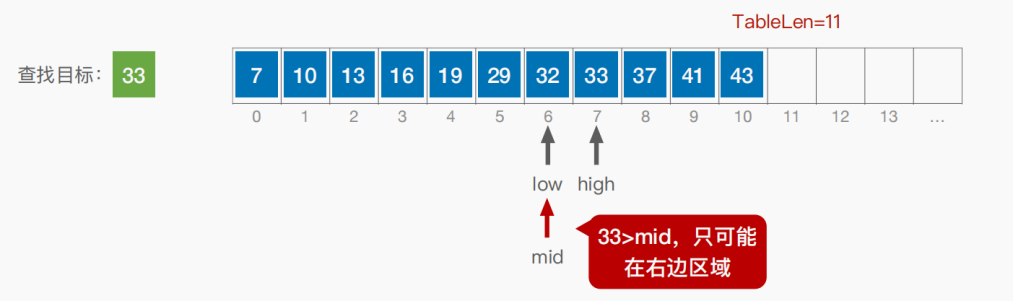
**折半查找**，⼜称“**⼆分查找**”，仅适⽤于**有序的顺序表**。

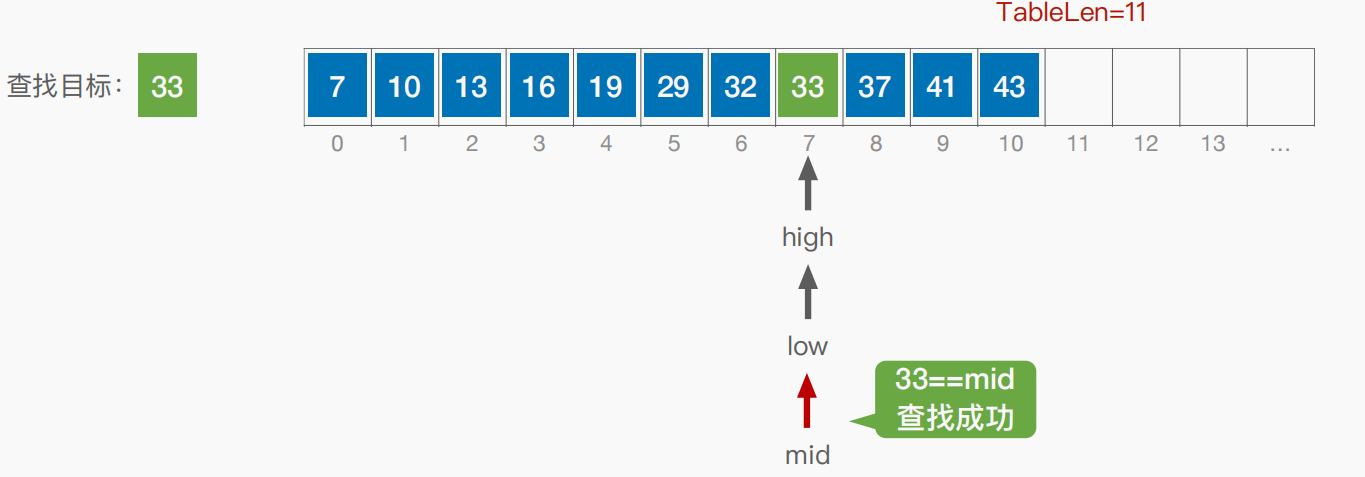
查找成功：



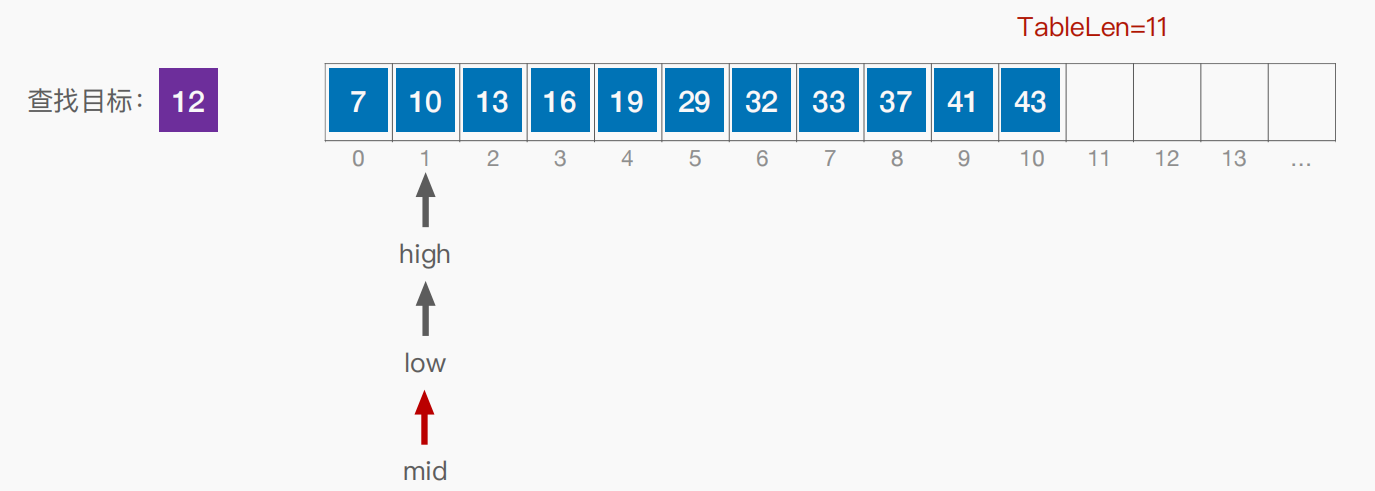


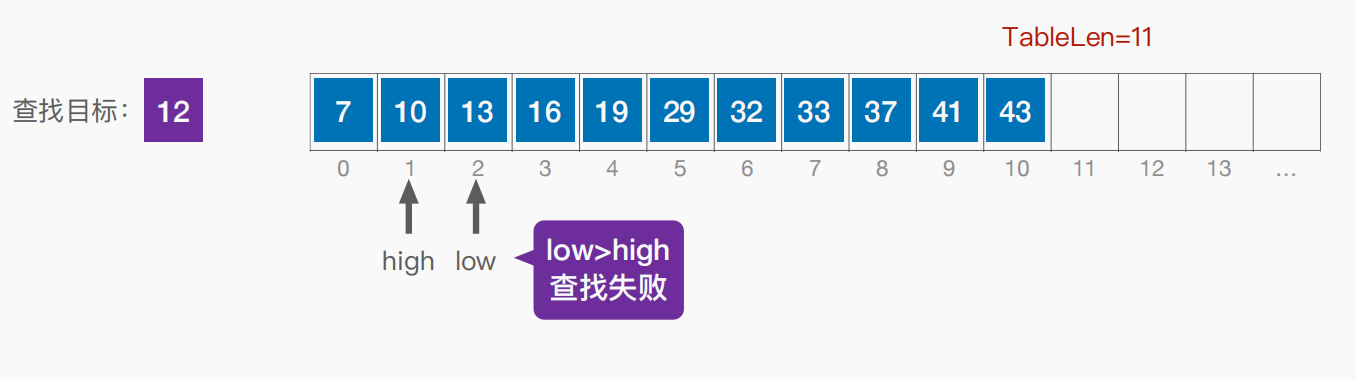




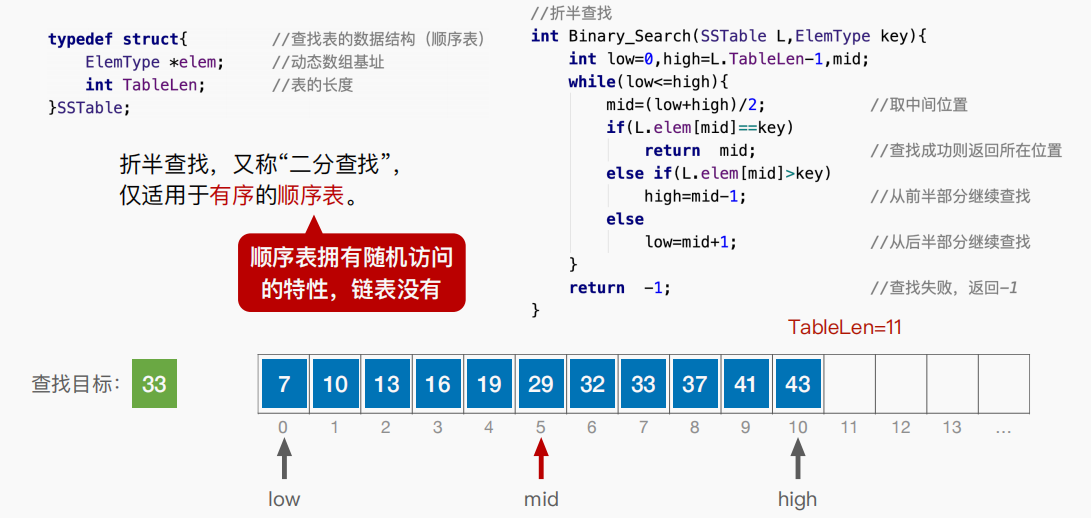
查找失败：



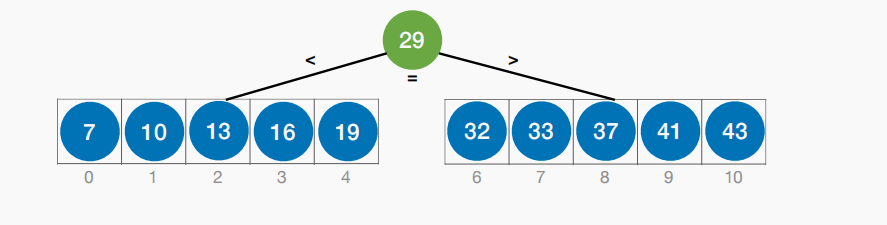


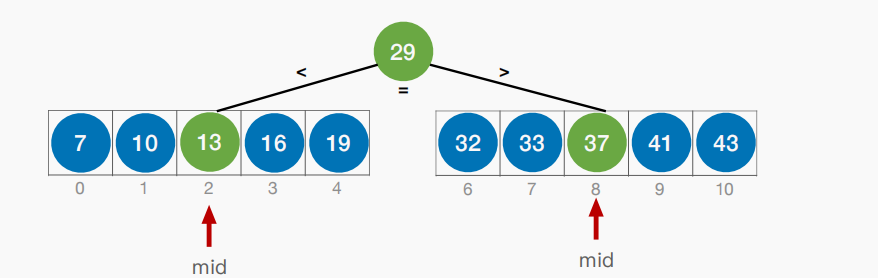
折半查找的实现：

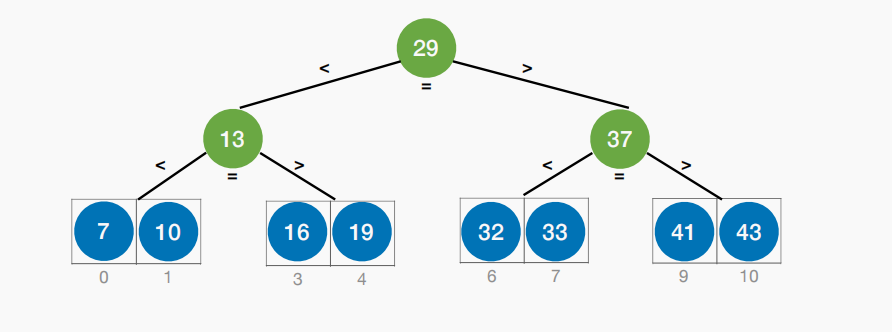
折半查找，⼜称“⼆分查找”，仅适⽤于**有序的顺序表**。

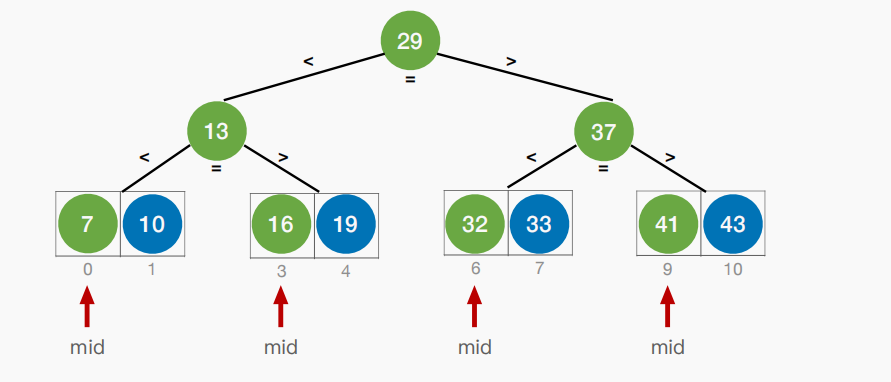


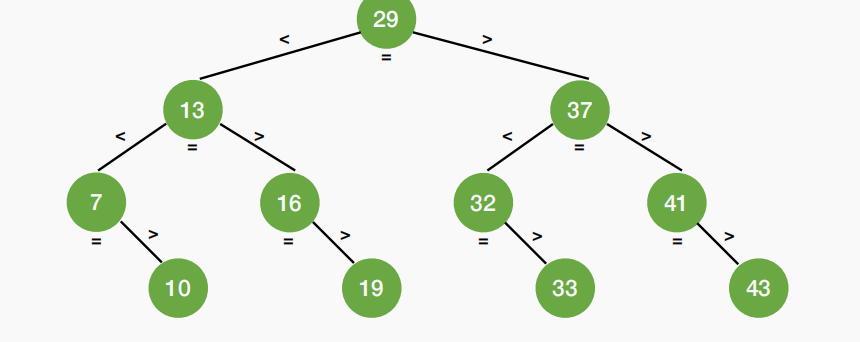
查找效率分析：

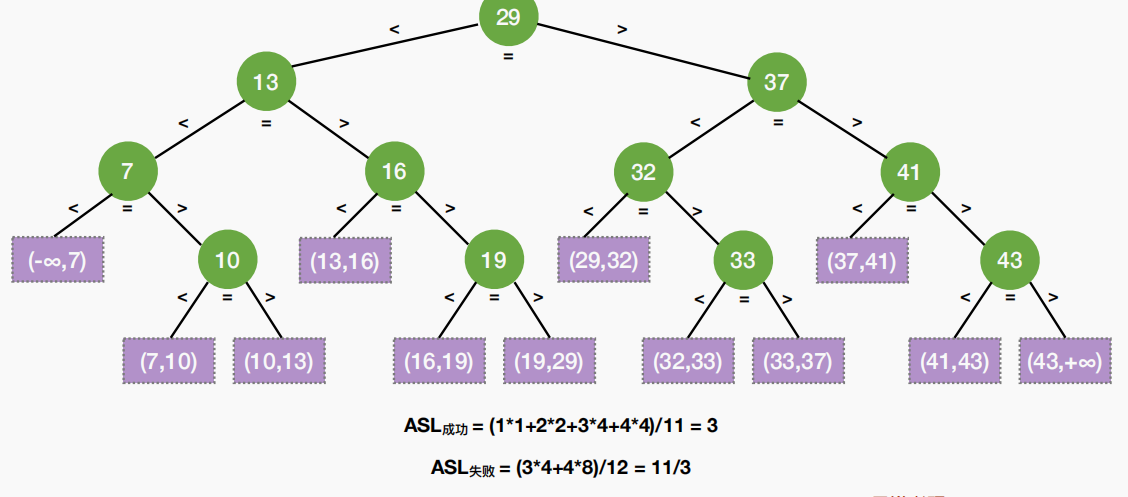




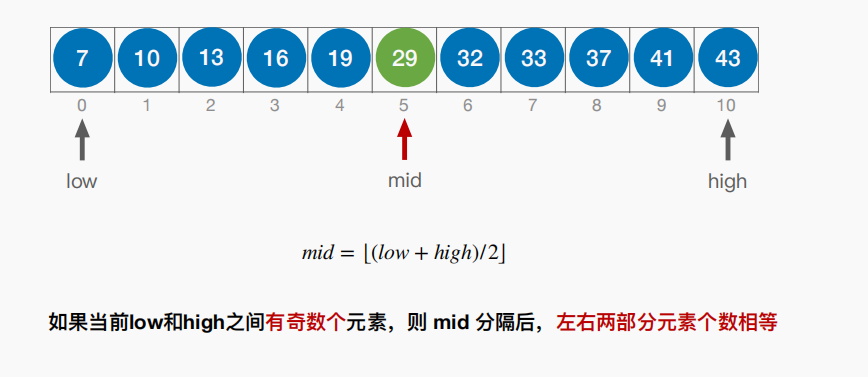




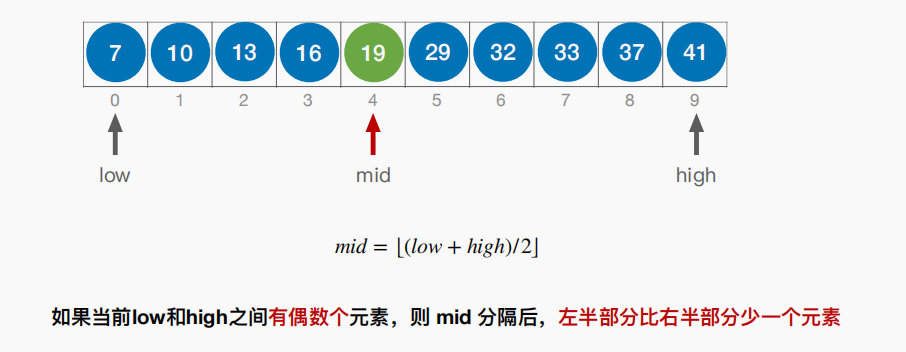




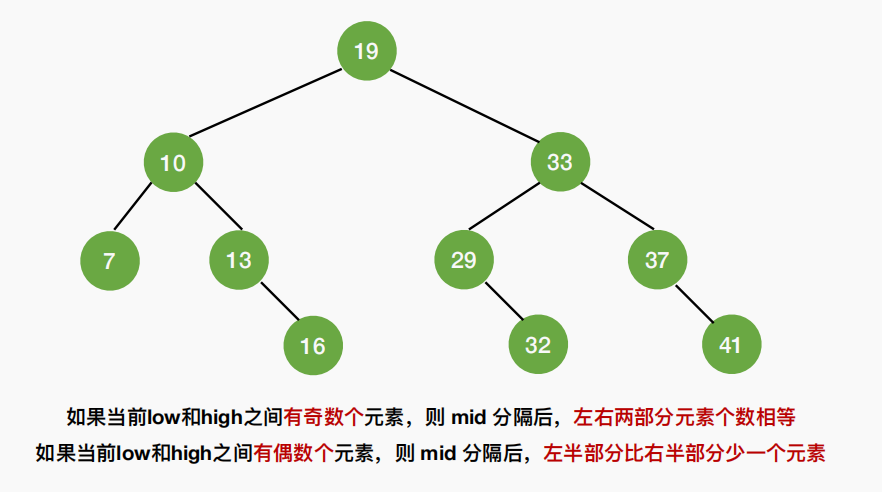
折半查找判定树的构造：

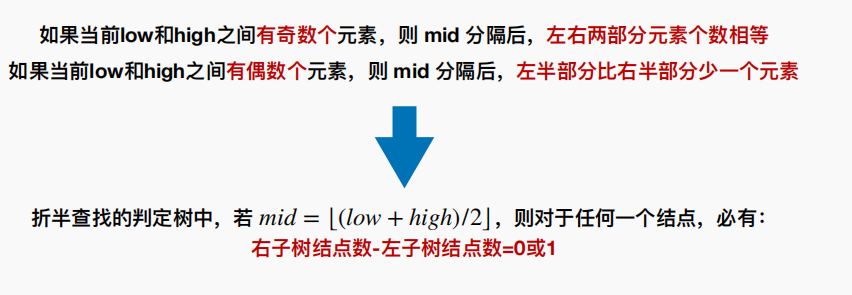


如果当前low和high之间有**奇数**个元素，则 mid 分隔后，**左右两部分元素个数相等**



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**

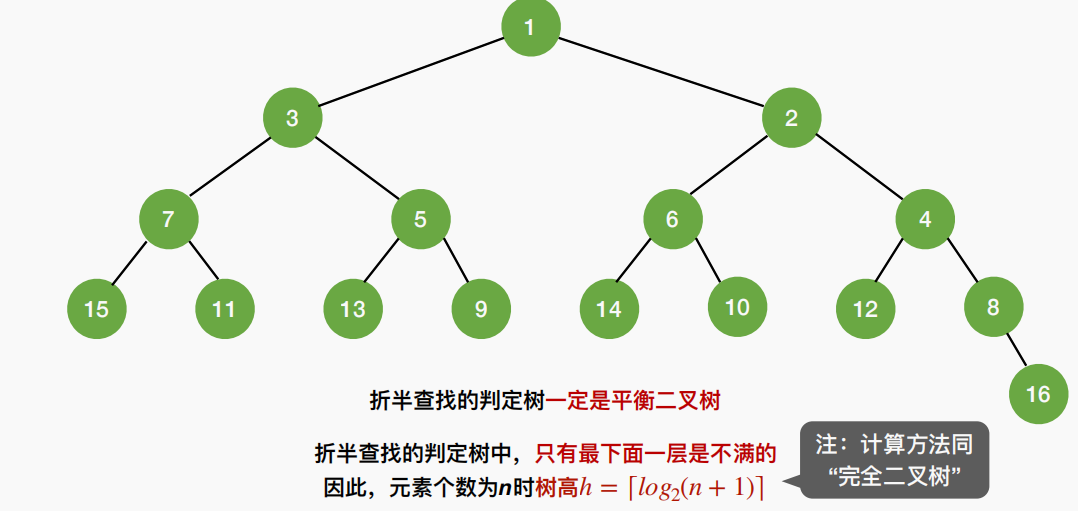


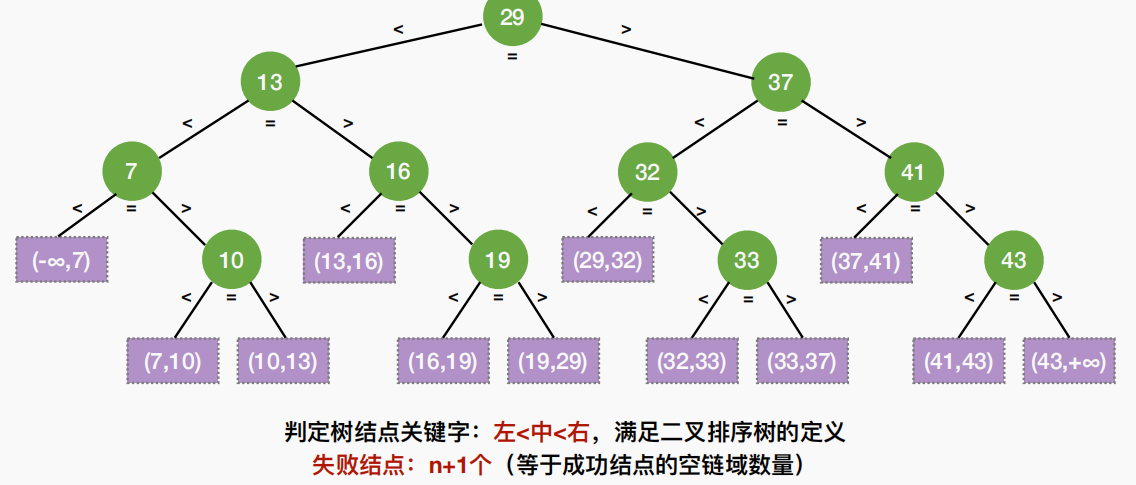


折半查找的判定树⼀定是**平衡⼆叉树**

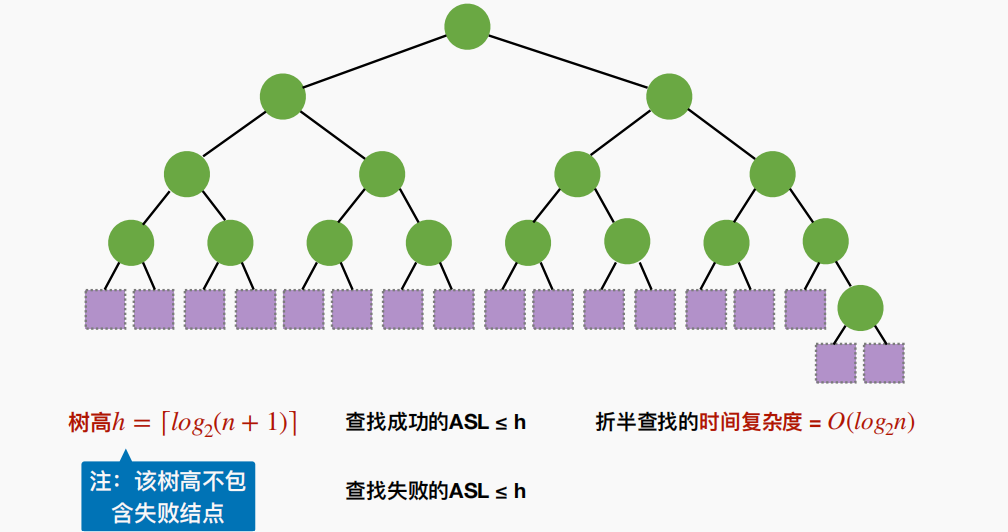
折半查找的判定树中，只有**最下⾯⼀层是不满**的

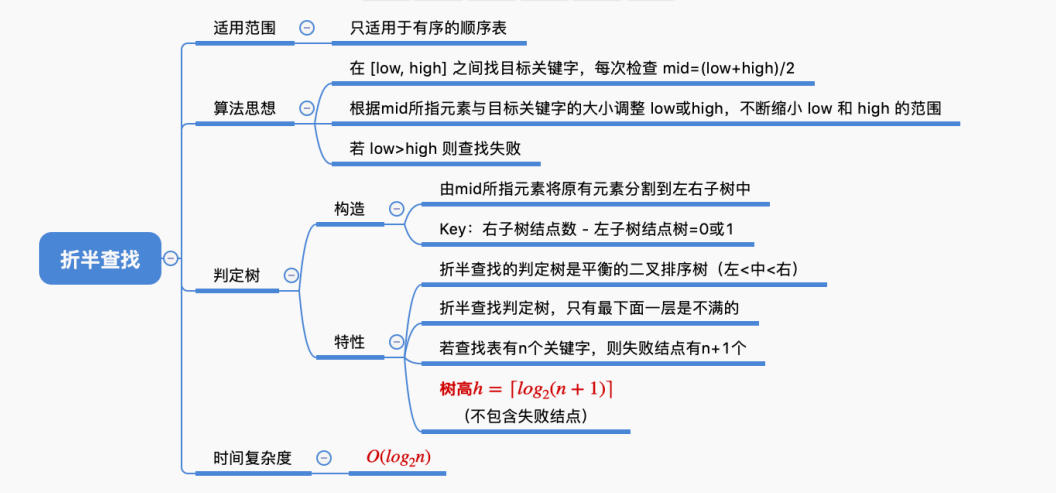
因此，元素个数为n时树⾼**h = ⌈log2(n + 1)⌉**



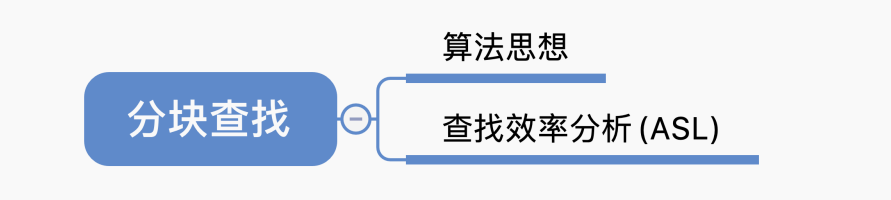


查找效率：

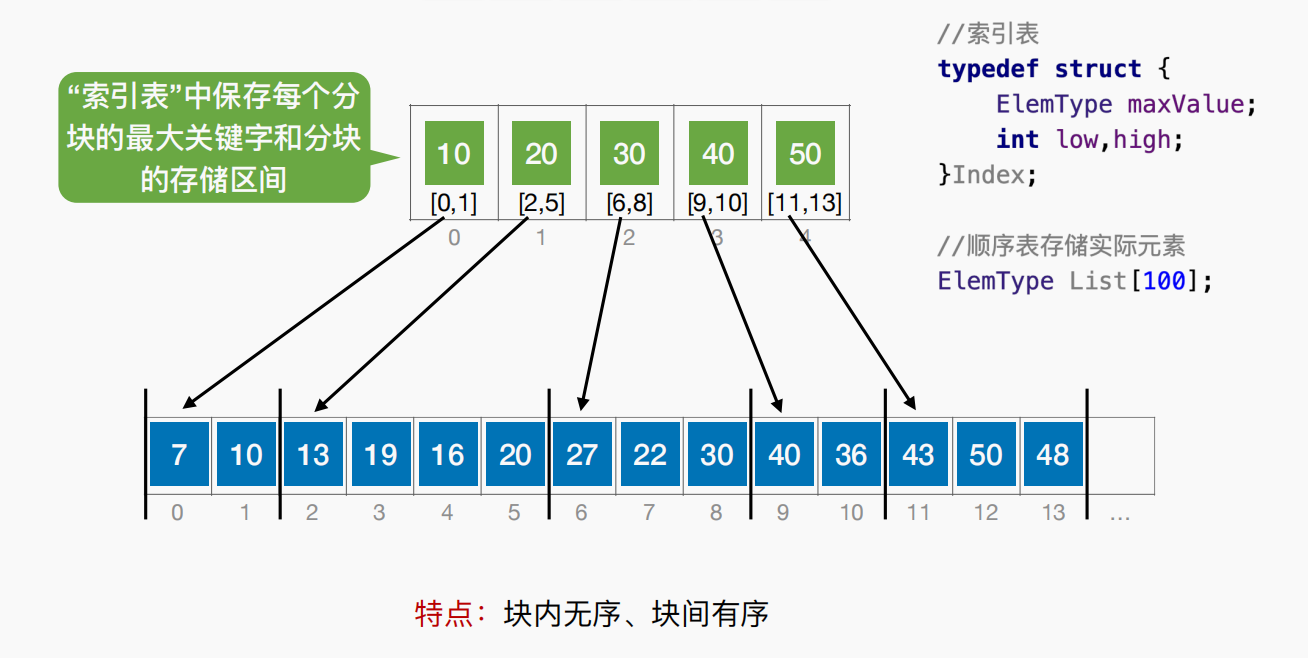


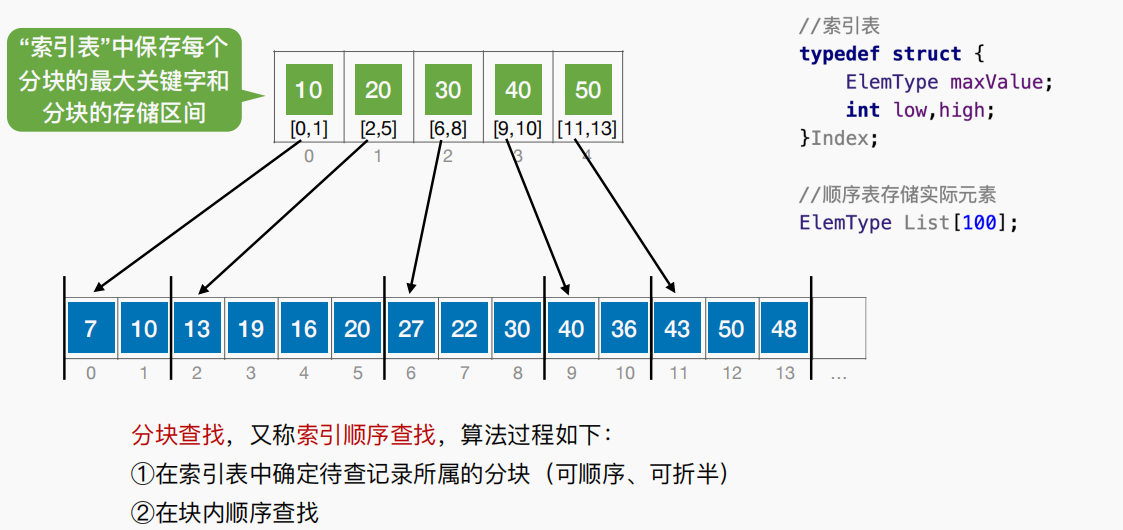


分块查找



分块查找的算法思想：



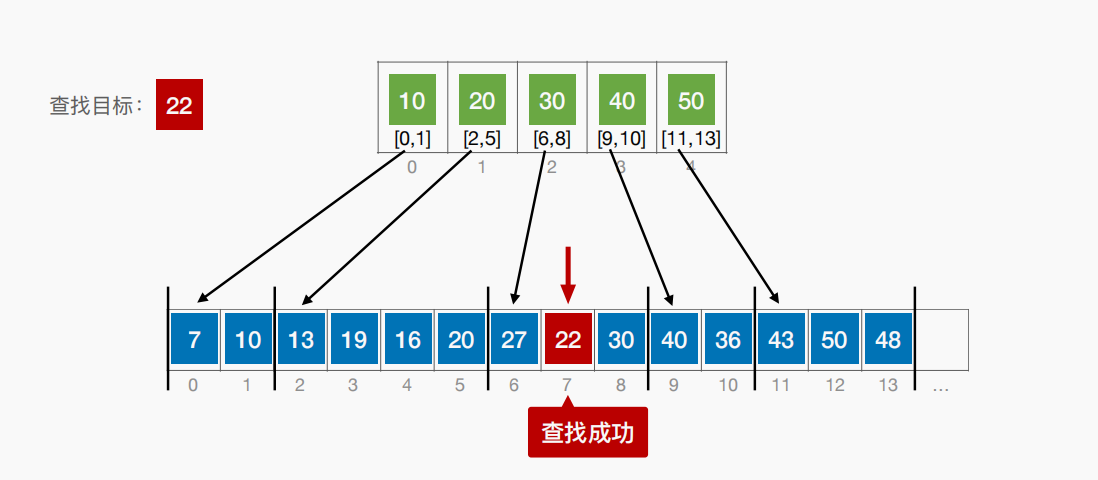


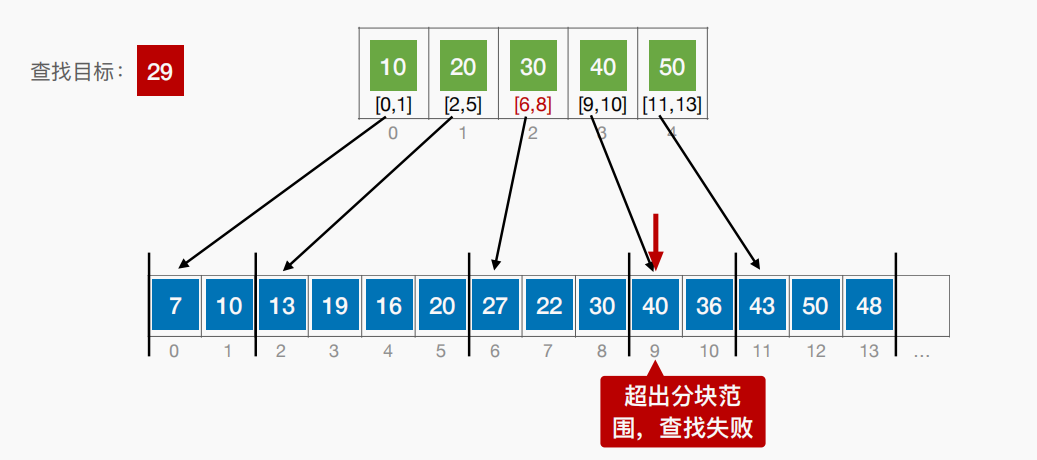
**分块查找**，⼜称**索引顺序查找**，算法过程如下：

①在索引表中确定待查**记录所属的分块**（可**顺序**、可**折半**）

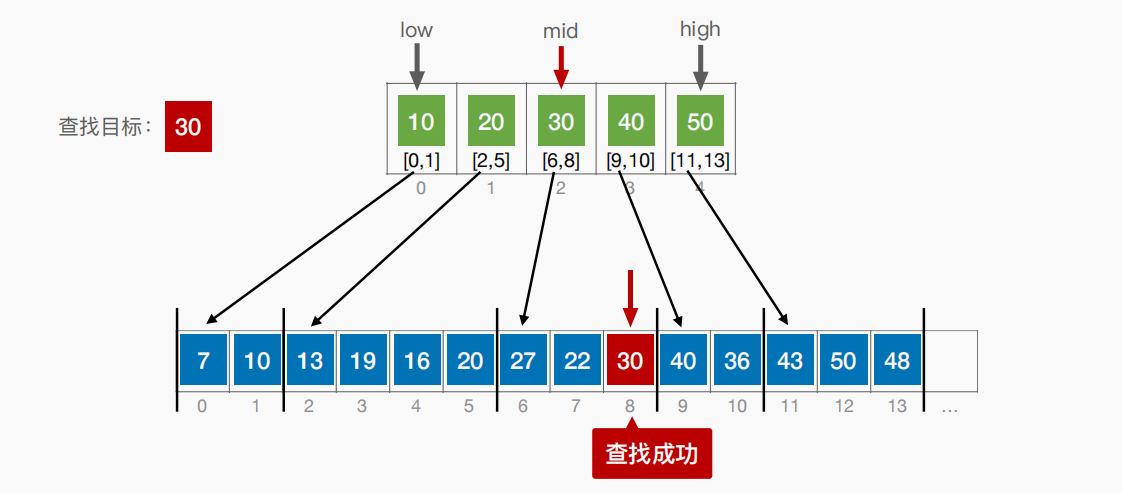
②在**块内顺序查找**

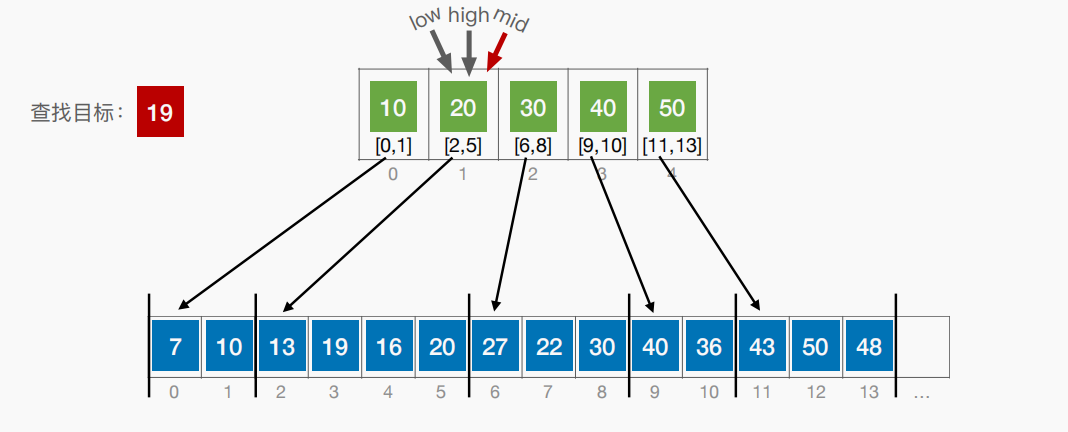
（1）⽤顺序查找查索引:

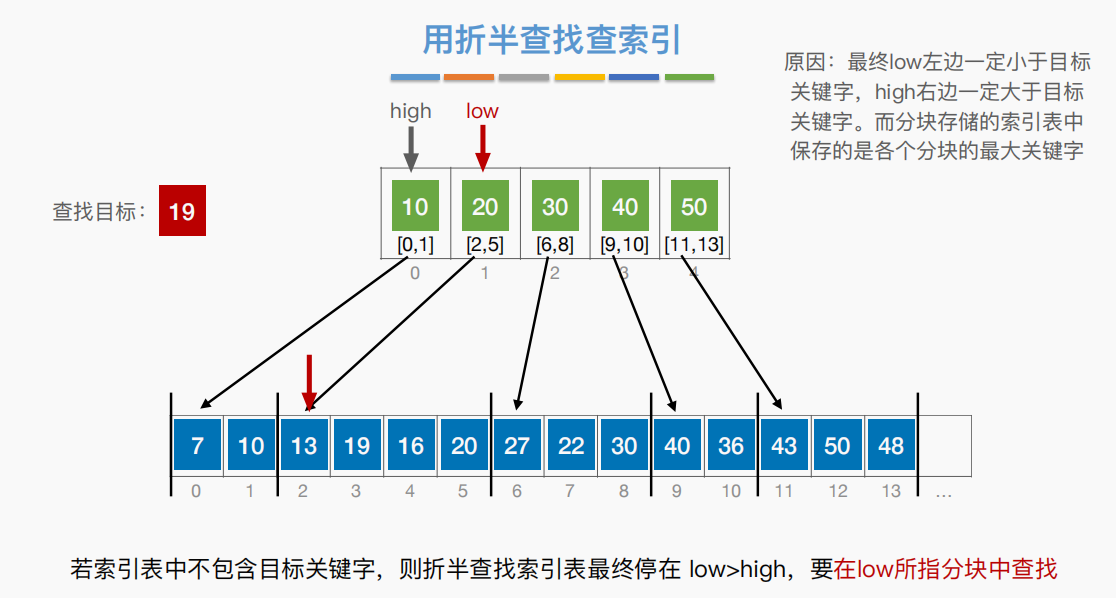




（2）⽤折半查找查索引:

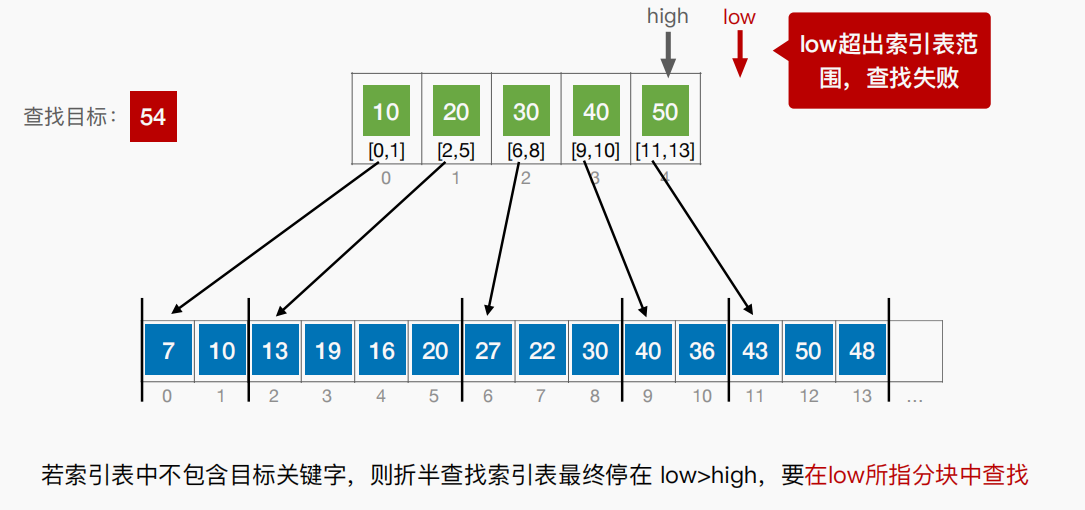




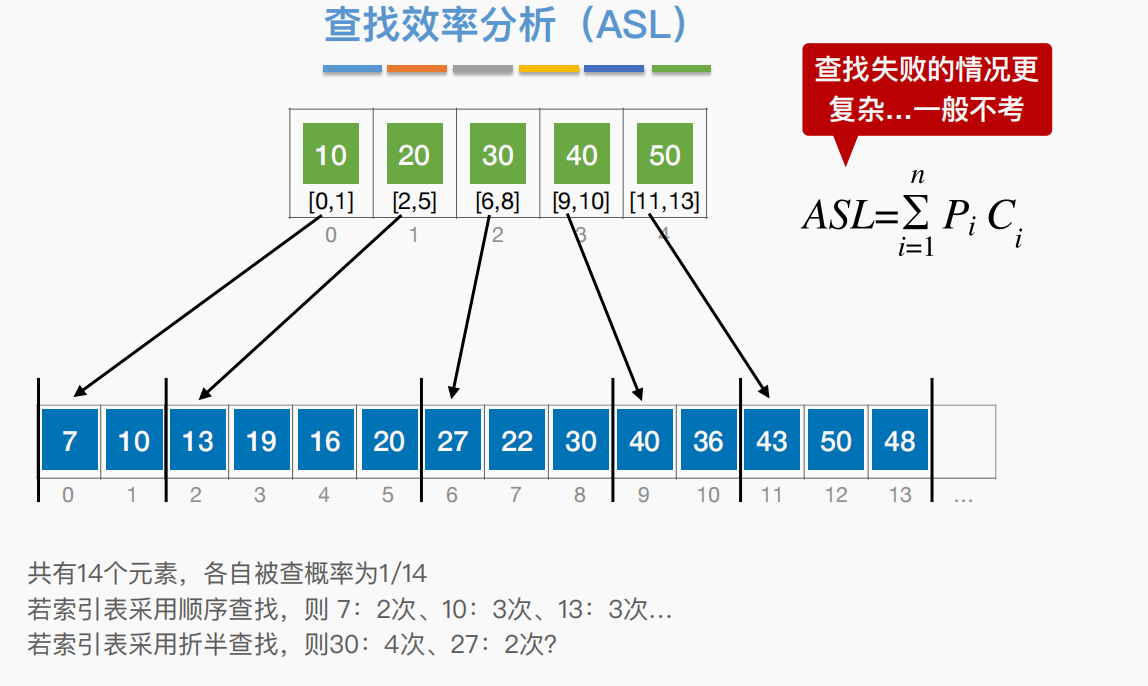


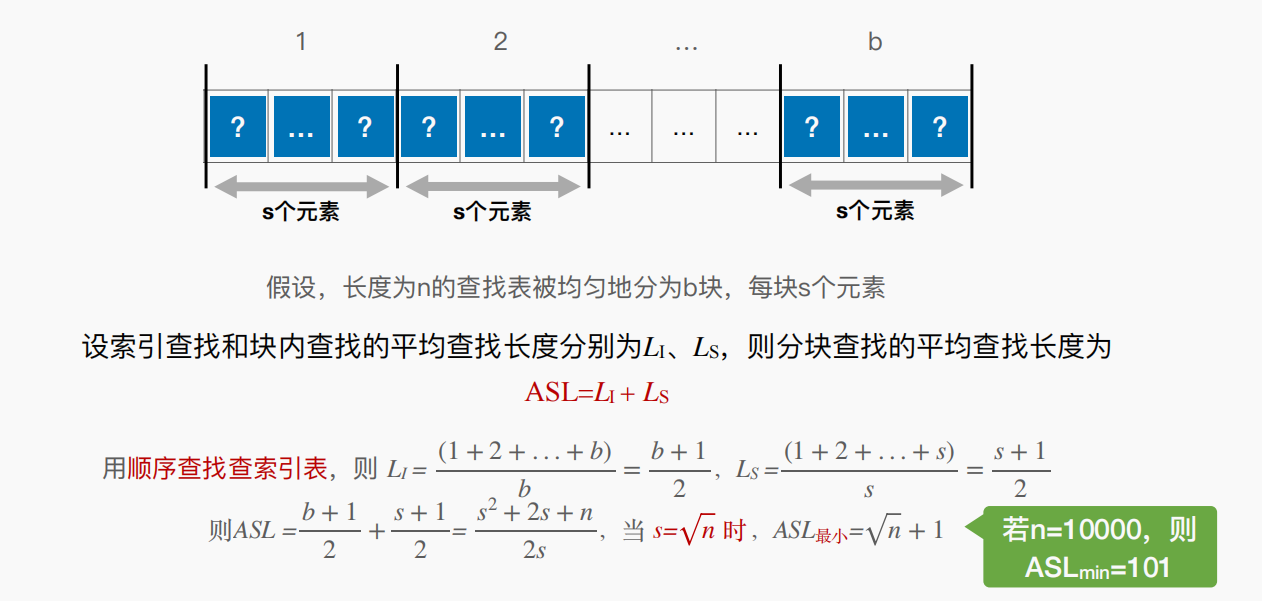
若索引表中**不包含⽬标关键字**，则折半查找索引表最终停在 low>high，要**在low所指分块中查找**

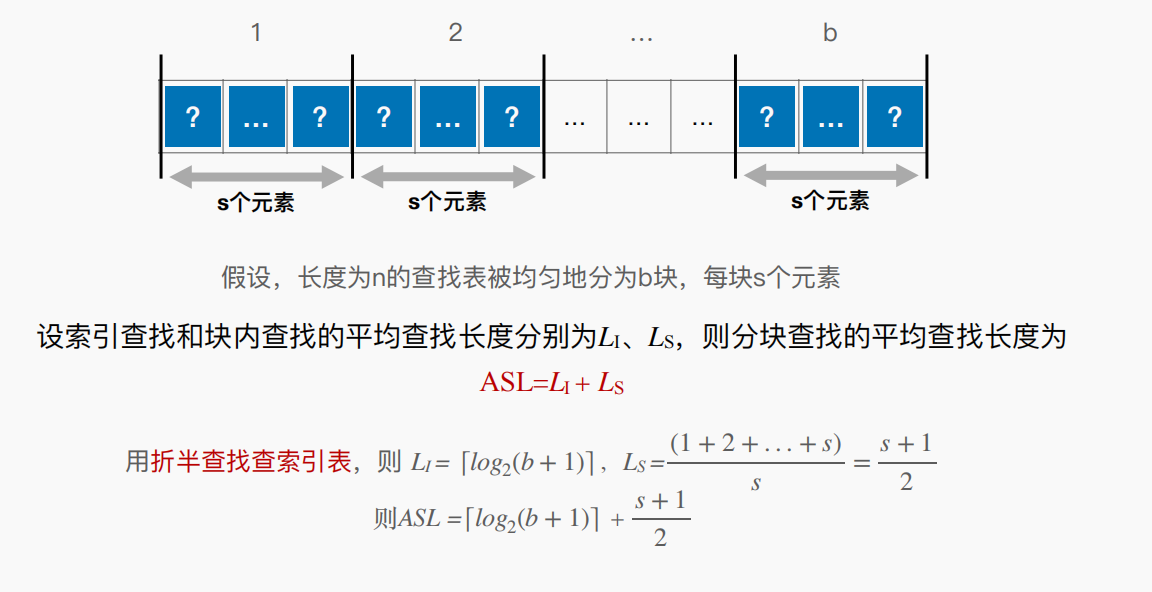
原因：最终low左边⼀定⼩于⽬标关键字，high右边⼀定⼤于⽬标关键字。⽽分块存储的索引表中保存的是各个分块的最⼤关键字

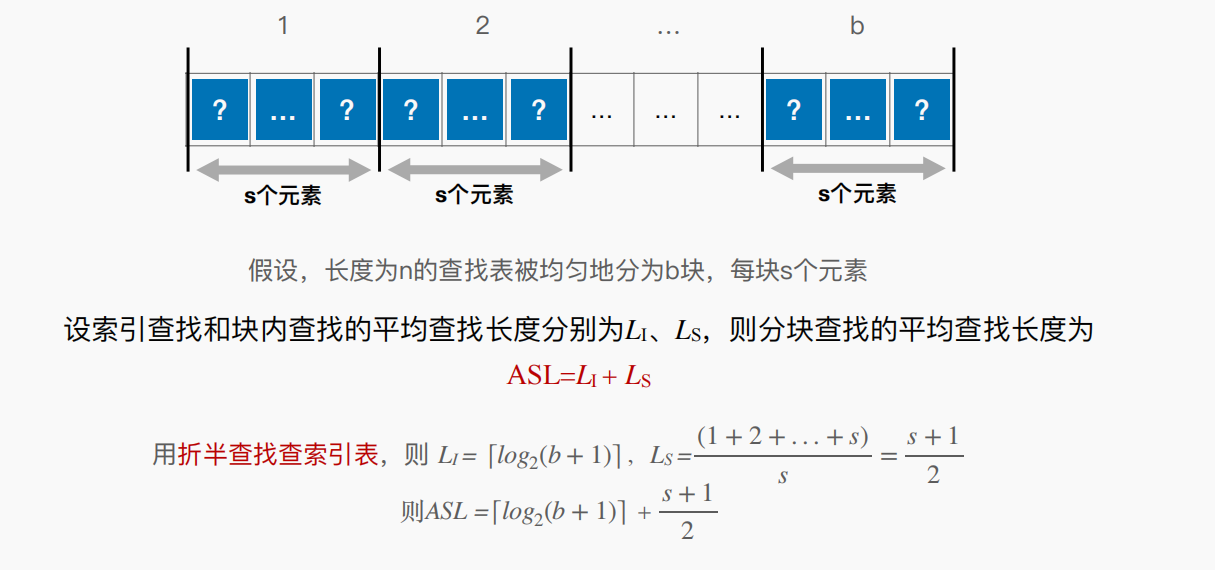


查找效率分析（ASL）：









总结：

