

## 9.1.2 折半查找

- 折半查找又称二分查找，它是一种效率很高的查找方法。
- 折半查找要求顺序表中元素是有序的，即表中元素按关键字有序。
- 假设有顺序表是递增有序的。

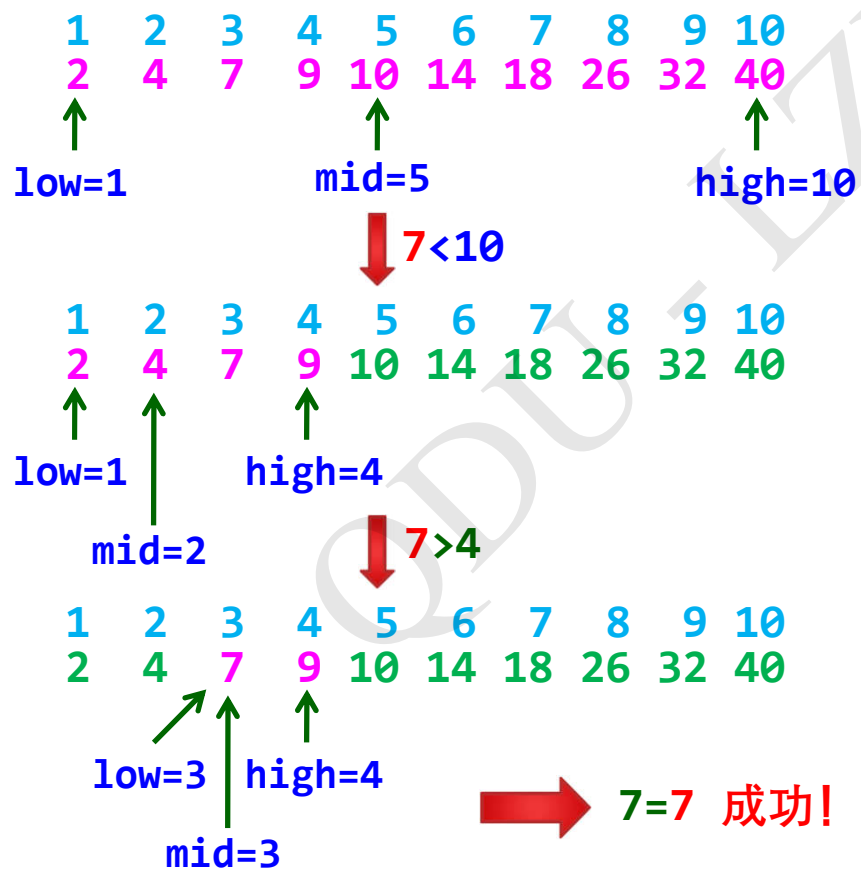
**基本思路：** 设 $elem[low..high]$ 是当前的查找区间，首先确定该区间的中点位置 $mid = \lfloor (low+high)/2 \rfloor$ ；然后将待查的 $key$ 值与 $elem[mid].key$ 比较：

- 若 $elem[mid].key = key$ ，则查找成功并返回该元素的逻辑序号；
- 若 $elem[mid].key > key$ ，则由表的有序性可知 $elem[mid..n].key$ 均大于 $key$ ，因此若表中存在关键字等于 $key$ 的元素，则该元素必定在位置 $mid$ 左子表 $elem[1..mid-1]$ 中，故新的查找区间是左子表 $elem[1..mid-1]$ ；
- 若 $elem[mid].key < key$ ，则要查找的 $key$ 必在位置 $mid$ 的右子表 $elem[mid+1..n]$ 中，即新的查找区间是右子表 $elem[mid+1..n]$ 。

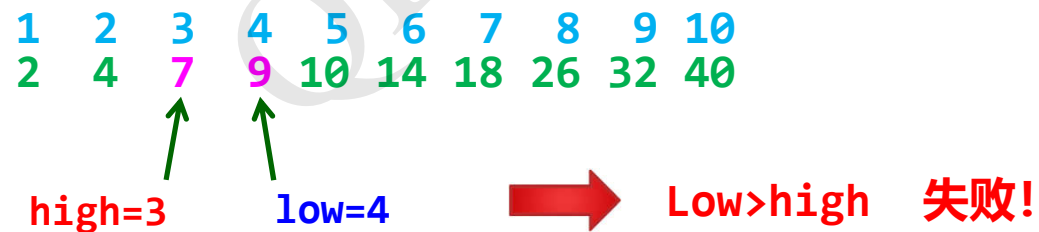
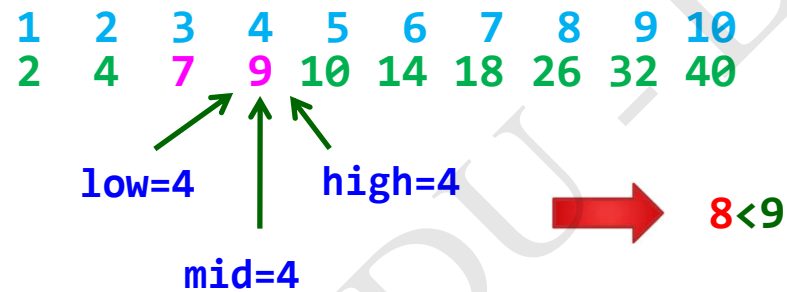
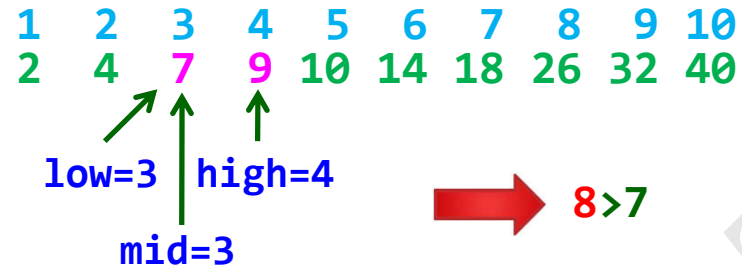
□ 下一次查找是针对新的子表重复上述的步骤。

【示例-1】在关键字有序序列 (2,4,7,9,10,14,18,26,32,40) 中采用折半查找方法查找关键字为7和8的元素。

解：查找7的折半查找过程如下：



解：查找8的折半查找过程如下(省略前两步骤)：



折半查找算法（在有序顺序表elem[1..n]中进行折半查找，成功时返回元素的逻辑序号，失败时返回0）：

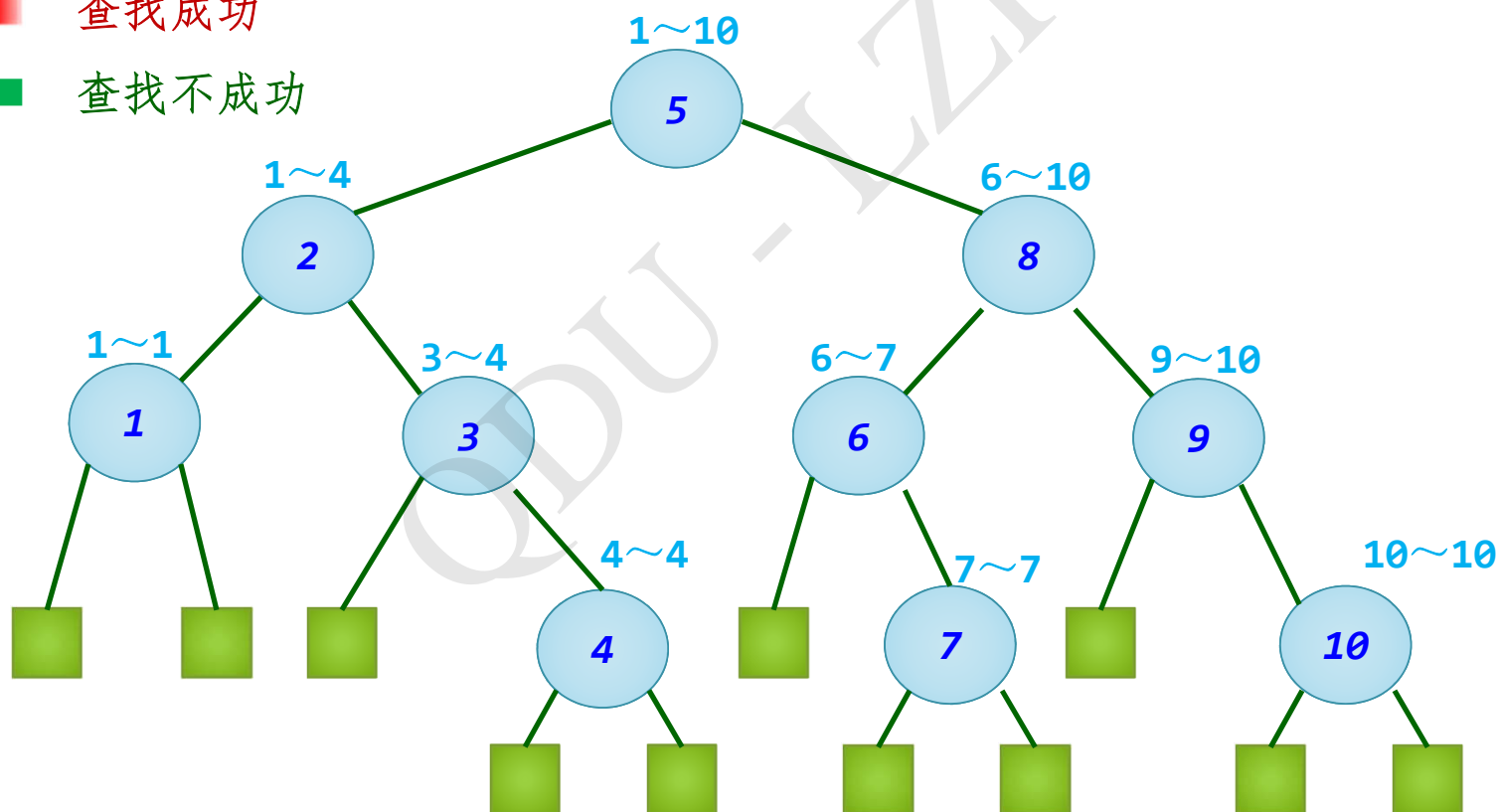
```
int Search_Bin(SSTable ST, KeyType key)
{ // 在有序表ST中折半查找其关键字等于key的数据元素。若找到，则返回
  // 该元素在表中的位置，否则返回0。算法9.2
  int low, high, mid;
  low = 1; // 置区间初值
  high = ST.length;
  while(low <= high) {
    mid = (low + high) / 2;
    if EQ(key, ST.elem[mid].key) // 找到待查元素
      return mid;
    else
      if LT(key, ST.elem[mid].key)
        high = mid - 1; // 继续在前半区间进行查找
      else
        low = mid + 1; // 继续在后半区间进行查找
  }
  return 0; // 顺序表中不存在待查元素
}
```

**算法分析：**折半查找过程构成一个判定树，把当前查找区间的中间位置上的记录作为根，左子表和右子表中的记录分别作为根的左子树和右子树。

$n=10$ 即 $\text{elem}[1..10]$ 的全部查找情况，圆圈中的数字代表下标值：

■ 查找成功

■ 查找不成功



当折半查找表中元素个数 $n$ 较大时, 可以将整个判定树近似看成是一棵满二叉树, 所有的叶子结点集中在同一层。不考虑外部结点, 二叉树的高度 $h = \lceil \log_2(n+1) \rceil$ 。所以有:

$$ASL_{\text{succ}} = \sum_{i=1}^n p_i C_i = \frac{1}{n} \sum_{i=1}^h i \times 2^{i-1} = \frac{n+1}{n} \log_2(n+1) - 1 \approx \log_2 n$$

$$ASL_{\text{unsucc}} = h = \lceil \log_2(n+1) \rceil$$

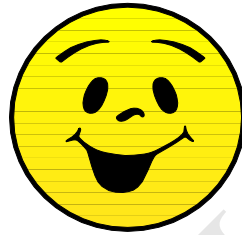
**【示例-2】** 已知一个长度为16的顺序表，其元素按关键字有序排序，若采用折半查找法查找一个不存在的元素，则比较的次数最多是（ ）。

A. 4      B. 5      C. 6      D. 7

说明：本题为2010年全国考研题。

$$h = \lceil \log_2(n+1) \rceil = h = \lceil \log_2 17 \rceil = 5$$





— END —