

检索效率：平均检索长度Average Search Length

查找或搜索是指从一组数据元素中找到需要的数据元素。衡量查找效率的主要标准是查找过程中平均比较次数，即平均检索长度ASL，定义如下：

$$ASL(n) = \sum_{i=1}^n p_i c_i$$

其中， n 是结点的个数， p_i 是查找第 i 个结点的概率，若不特别声明，一般认为每个结点的查找概率是相等的，即 $p_1 = p_2 = \cdots = p_n = 1/n$ 。

c_i 是查找成功到第 i 个结点所需要比较的次数。

uc_i 是查找失败第 i 个结点所需要比较的次数。

$$ASL_{\text{成功}} = \frac{1}{n} \sum_{i=1}^n c_i$$

$$ASL_{\text{失败}} = \frac{1}{n} \sum_{i=1}^n uc_i$$

二分查找ASL

对**有序表** (5, 10, **25**, 27, 30, 35, **45**, 49, 50, **52**, 55, 60, 70)
进行**二分查找**

比较1次成功的是45 (1个)

比较2次成功的是25, 52 (2个)

比较3次成功的是5, 30, 49, 60 (4个)

比较4次成功的是10, 27, 36, 50, 55, 70 (6个)

$$ASL = \frac{1 \times 1 + 2 \times 2 + 3 \times 4 + 4 \times 6}{13} = \frac{41}{13}$$

二分查找的效率 (ASL)

推导过程

1次比较就查找成功的元素有1个 (2^0) , 即中间值;

2次比较就查找成功的元素有2个 (2^1) , 即1/4处 (或3/4) 处;

3次比较就查找成功的元素有4个 (2^2) , 即1/8处 (或3/8) 处...

4次比较就查找成功的元素有8个 (2^3) , 即1/16处 (或3/16) 处...

.....

则第j次比较时查找成功的元素会有 (2^{j-1}) 个;

为方便起见, 假设表中全部n个元素 = $2^j - 1$ 个, 此时就不讨论第j次比较后还有剩余元素的情况了。

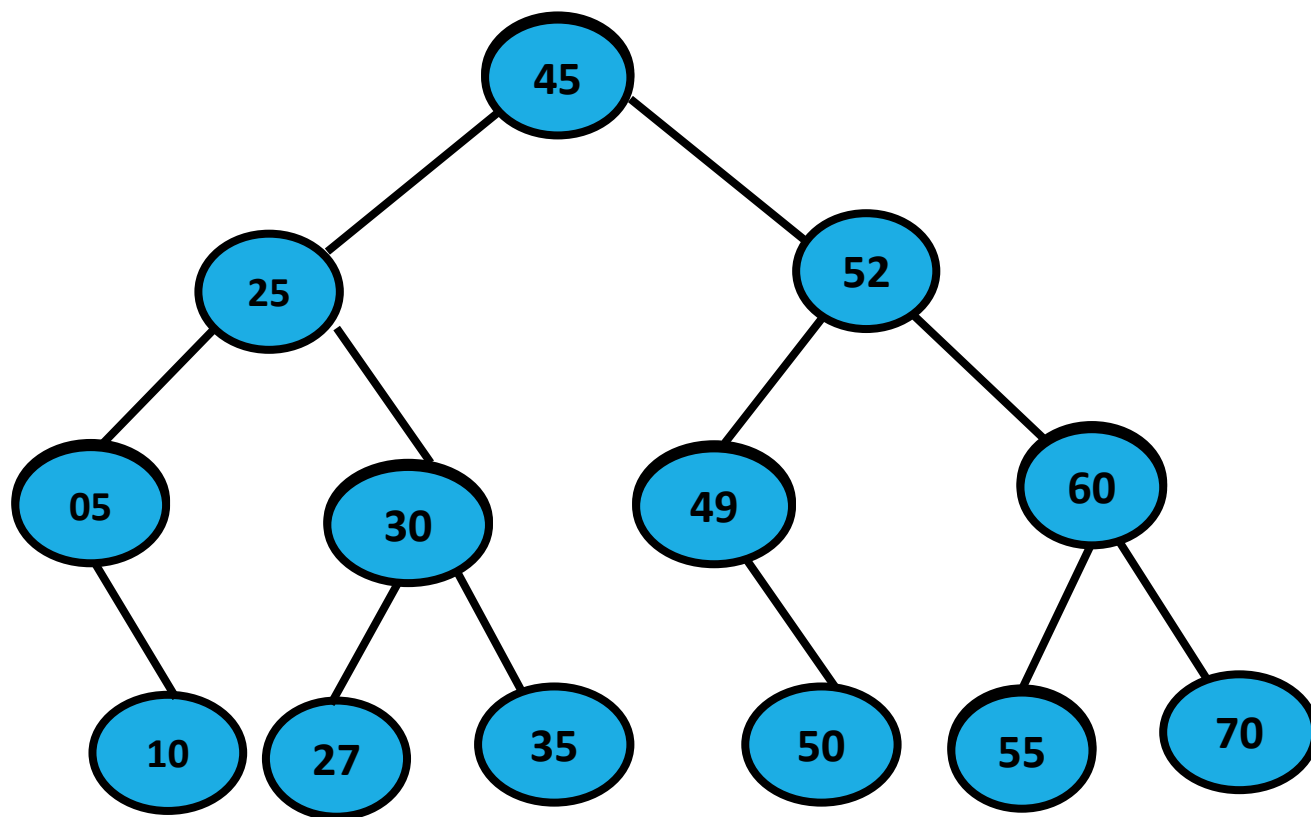
$$\text{全部比较总次数} = 1 \times 2^0 + 2 \times 2^1 + 3 \times 2^2 + 4 \times 2^3 \dots + j \times 2^{j-1} = \sum_{i=1}^j i \times 2^{i-1}$$

$$ASL = \frac{1}{n} \times \sum_{i=1}^j i \times 2^{j-1} = \frac{n+1}{n} \times \log_2^{(n+1)} - 1 \approx \log_2^n$$

二分查找判定树

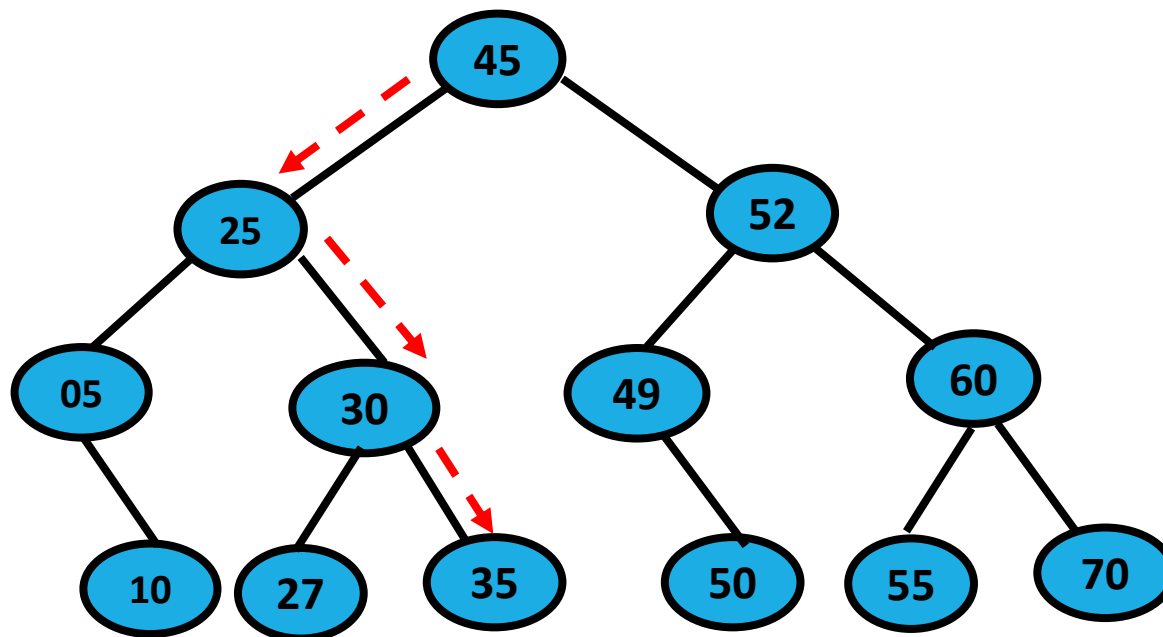
对有序表 (5, 10, 25, 27, 30, 35, 45, 49, 50, 52, 55, 60, 70) 进行二分查找

一次比较查找成功的元素作为二叉树的根，二次比较查找成功的元素是前半区间和后半区间中间的元素，分别作为根的左右子树，依次类推，由此得到的二叉树称为**二分查找判定树**

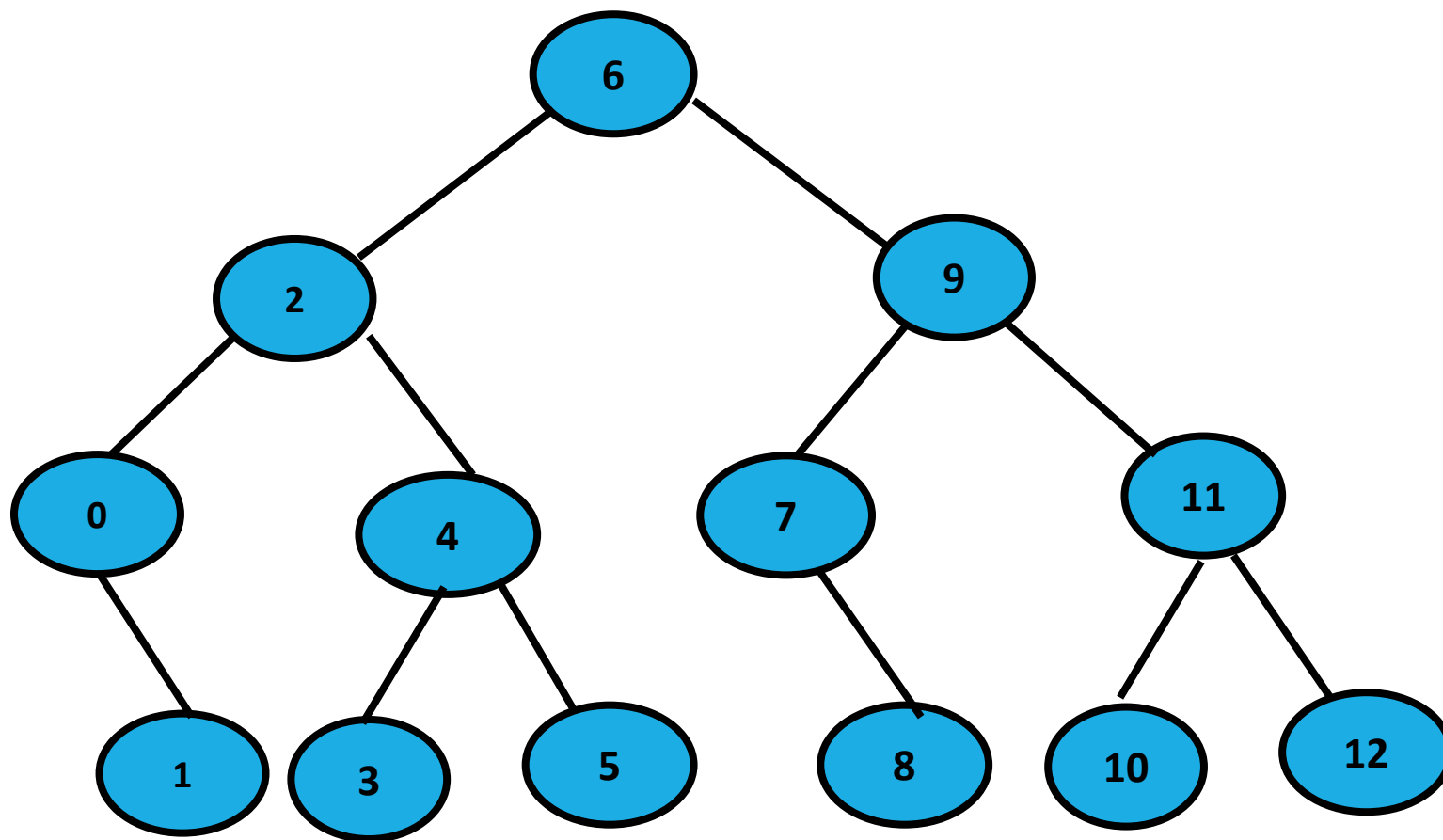


二分查找判定树

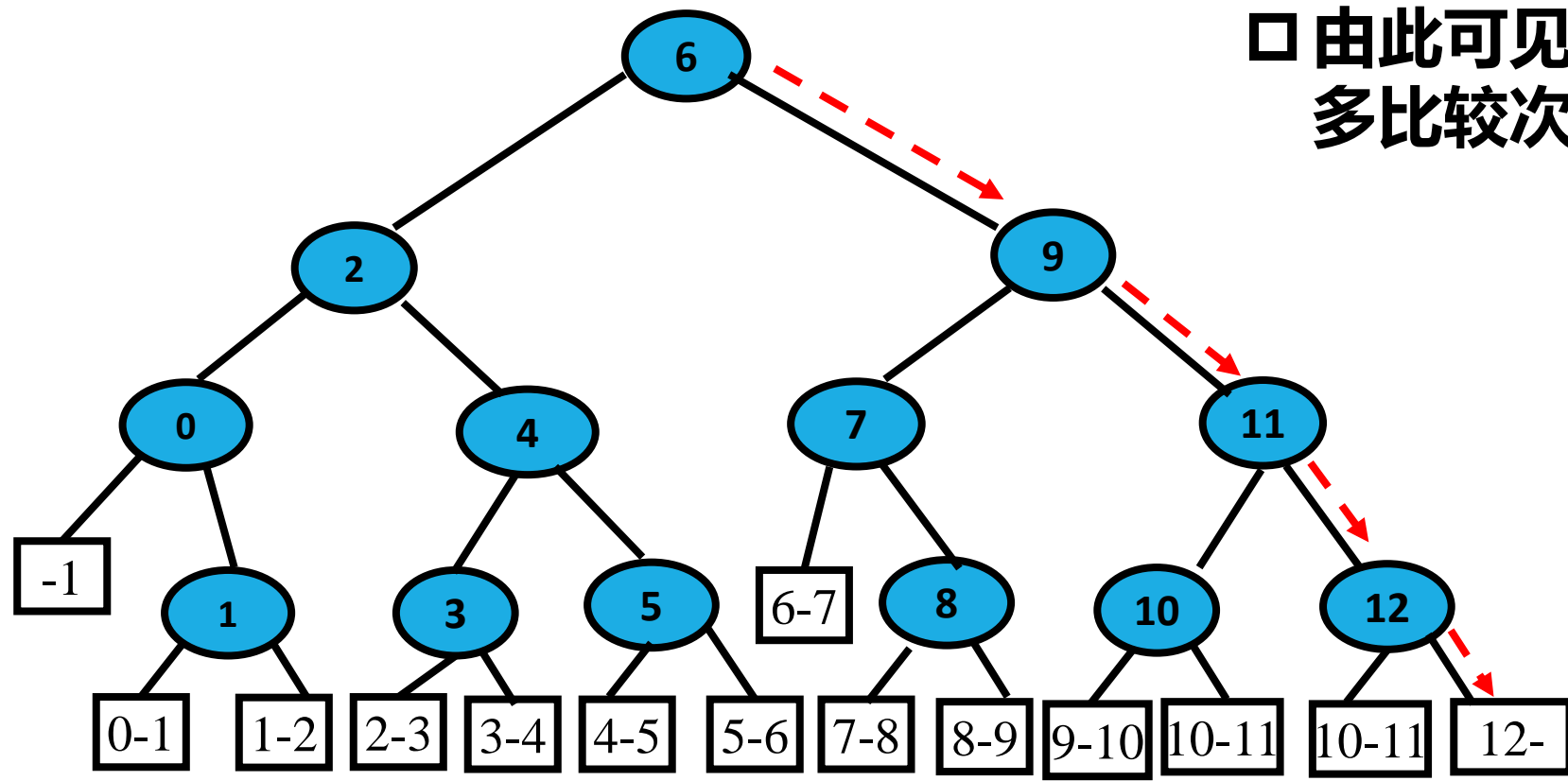
- ◆ **查找成功**是查找二分查找判定树中已有的结点，比较从判定树的根结点开始，走了一条从根结点到该结点的路径。路径上经过的结点个数即为查找比较的次数。
- ◆ **查找失败**的过程走了一条从根结点到其扩充二叉树的外部结点的路径



◆ 对长度为**13**的任意**有序表**进行**二分查找**，得到的二分查找判定树**形态是相同的**



□ 查找成功时进行比较的次数最多不超过该树的深度。而具有 n 个结点的判定树的深度为 $\lfloor \log_2 n \rfloor$ 。所以折半查找法在查找成功时的比较次数最多为 $\lfloor \log_2 n \rfloor + 1$ 。如果考虑到查找不成功的情况，则判定树如下所示（方框表示查找不成功的情况）：



□ 由此可见，查找不成功时的最多比较次数也是 $\lfloor \log_2 n \rfloor + 1$ 。