

9.2 线性表的查找

线性表查找的主要方法有：

(1) 顺序查找

(2) 二分查找

(3) 分块查找

线性表有顺序和链式两种存储结构。这里介绍以顺序表作为存储结构时实现线性表的查找算法。定义被查找的顺序表类型定义如下：

```
#define MAXL <表中最多记录个数>

typedef struct
{
    KeyType key;           //KeyType为关键字的数据类型
    InfoType data;        //其他数据项
} NodeType;

typedef NodeType SeqList[MAXL]; //查找顺序表类型
```



静态查找表

9.2.1 顺序查找

思路：从表的一端开始，顺序扫描线性表，依次将扫描到的关键字和给定值 k 相比较：



若当前扫描到的关键字与 k 相等，则查找成功；若扫描结束后，仍未找到关键字等于 k 的记录，则查找失败。

顺序查找的算法如下（在顺序表 $R[0..n-1]$ 中查找关键字为 k 的元素，成功时返回找到的元素的逻辑序号，失败时返回0）：

```
int SeqSearch(SeqList R,int n,KeyType k)
{   int i=0;
    while (i<n && R[i].key!=k)    //从表头往后找
        i++;
    if (i>=n)                        //未找到返回0
        return 0;
    else
        return i+1;                //找到返回逻辑序号i+1
}
```

❶ 成功情况下的平均查找长度ASL

查找到表中第*i*个记录*R[i-1]*时，需比较*i*次。因此成功时的顺序查找的平均查找长度为：

$$ASL_{sq} = \sum_{i=1}^n p_i c_i = \frac{1}{n} \sum_{i=1}^n i = \frac{1}{n} \times \frac{n(n+1)}{2} = \frac{n+1}{2}$$

查找成功时的平均比较次数约为表长的一半。

② 不成功情况下的平均查找长度ASL

查找不成功时需要和表中所有元素都比较一次，所以，不成功时的平均查找长度为 n 。

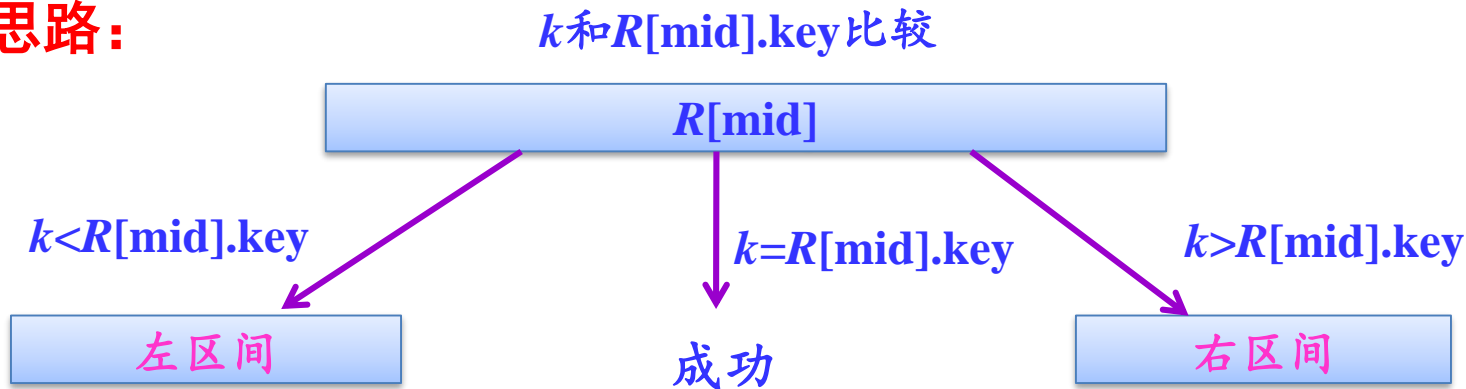


顺序查找的时间复杂度为 $O(n)$ 。

9.2.2 折半查找

折半查找也称为二分查找，要求线性表中的记录必须已按关键字值有序（递增或递减）排列。

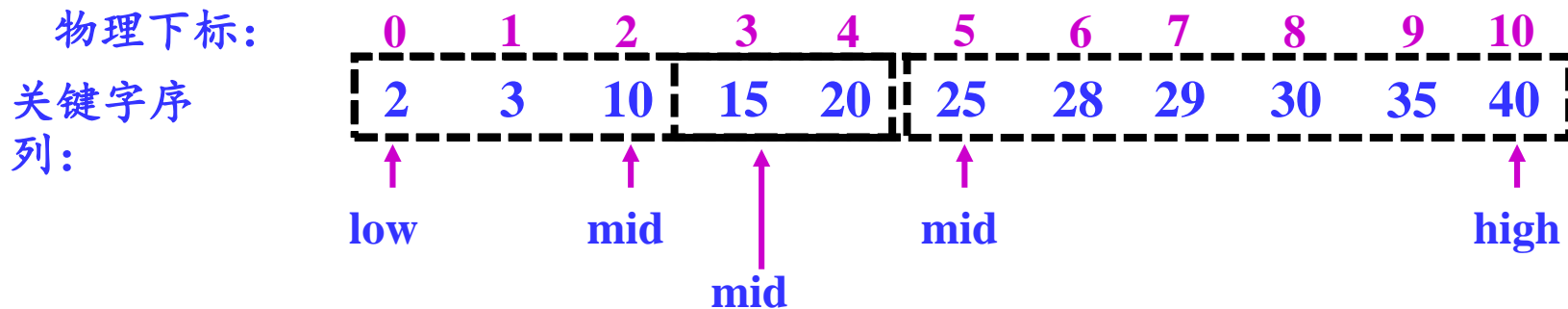
思路：



折半查找演示

例如，在关键字有序序列{2,3,10,15,20,25,28,29,30,35,40}中采用折半查找法查找关键字为15的元素。

找关键字为15的记录



查找成功，关键字为15的记录的逻辑序号为4

关键字比较次数为3

算法如下（在有序表 $R[0..n-1]$ 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 BinSearch(SeqList R,int n,KeyType k)
{
    int low=0,high=n-1,mid;
    while (low<=high)                //当前区间存在元素时循环
    {
        mid=(low+high)/2;
        if (R[mid].key==k)           //查找成功返回其逻辑序号mid+1
            return mid+1;
        if (k<R[mid].key)            //继续在R[low..mid-1]中查找
            high=mid-1;
        else                          //继续在R[mid+1..high]中查找
            low=mid+1;
    }
    return 0;
}
```

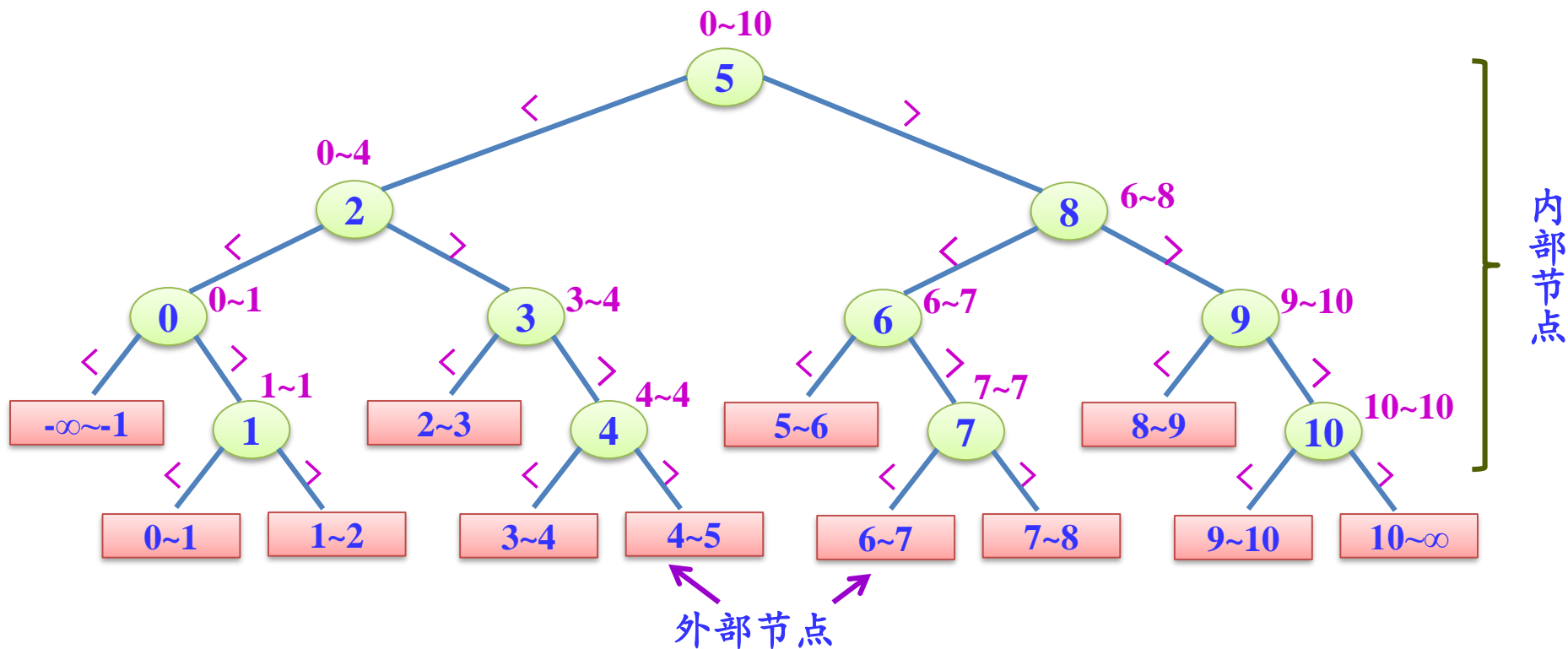
思考题

折半查找可以设计成递归算法，如何实现？

二分查找过程可用二叉树来描述：

- 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。

这样的二叉树称为判定树或比较树。



$R[0..10]$ 的二分查找的判定树 ($n=11$)

外部节点即查找失败对应的节点，是虚拟的

n 个关键字：内部节点为 n 个，外部节点为 $n+1$ 个

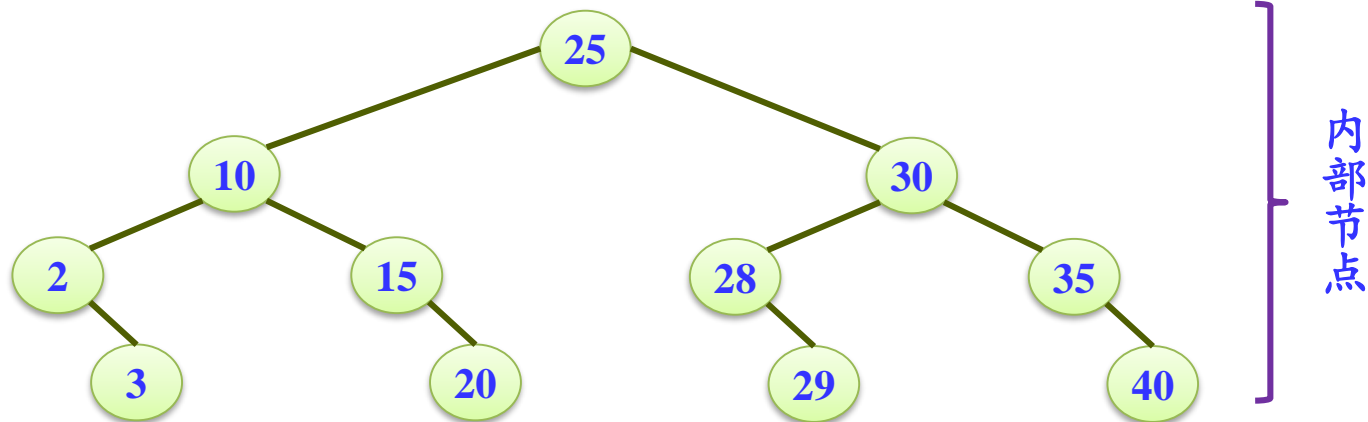
【例 9-1】 对于给定 11 个数据元素的有序表 {2,3,10,15,20,25,28,29,30,35,40}, 采用二分查找, 试问:

(1) 若查找给定值为 20 的元素, 将依次与表中哪些元素比较?

(2) 若查找给定值为 26 的元素, 将依次与哪些元素比较?

(3) 假设查找表中每个元素的概率相同, 求查找成功时的平均查找长度和查找不成功时的平均查找长度。

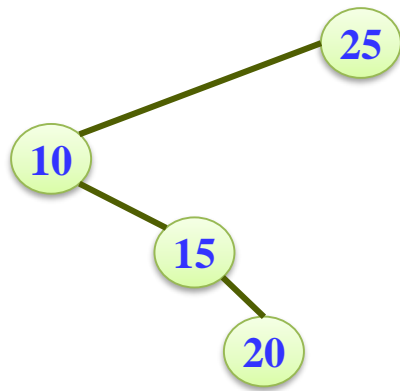
判定树：



解：（1）若查找给定值为**20**的元素，依次与表中25、10、15、20元素比较，共比较4次。

成功二分查找：恰好是走了一条从判定树的根到被查记录的路径，经历比较的关键字次数恰为该记录在树中的层数。

查找20

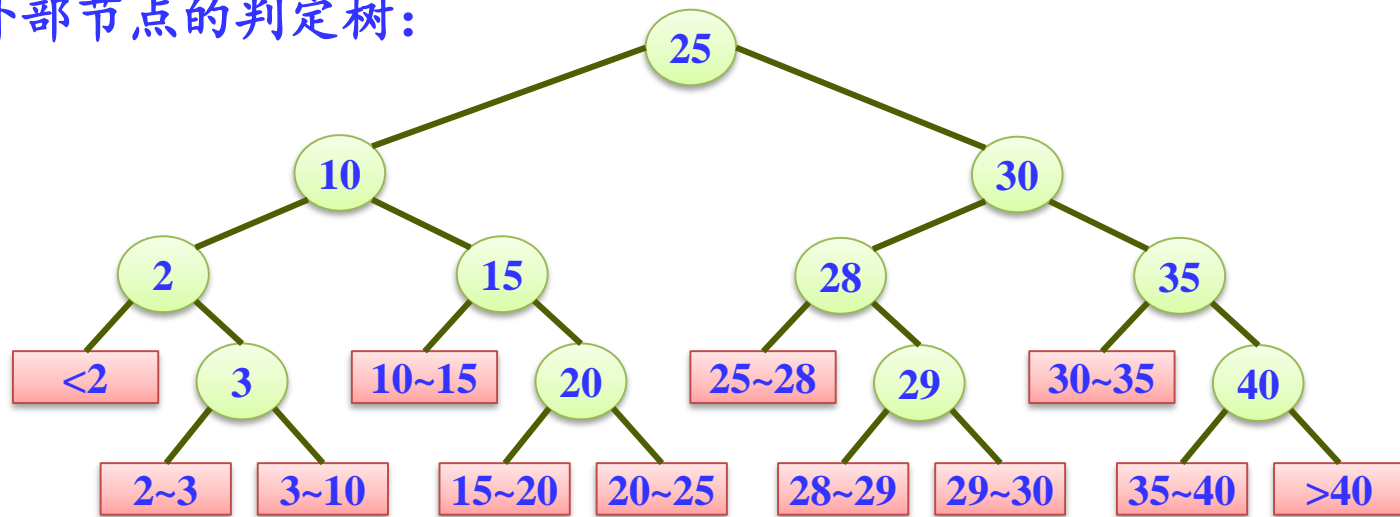


关键字比较4次

(3) 在查找成功时，会找到图中某个内部节点，则成功时的平均查找长度：

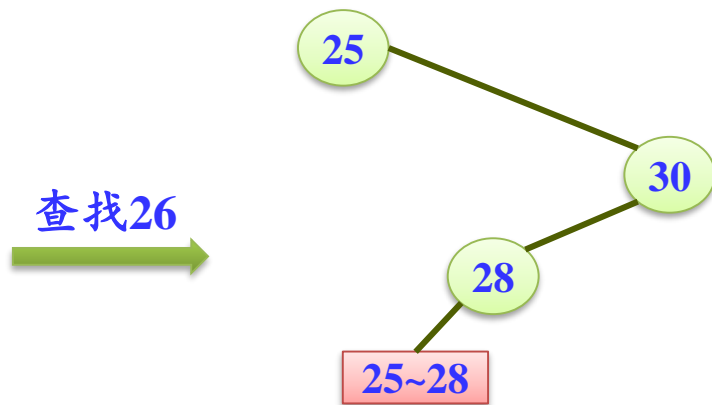
$$ASL_{\text{成功}} = \frac{1 \times 1 + 2 \times 2 + 4 \times 3 + 4 \times 4}{11} = 3$$

带外部节点的判定树：



(2) 若查找给定值为**26**的元素，依次与25、30、28元素比较，共比较3次。

不成功二分查找：比较过程经历了一条从判定树根到某个外部节点的路径，所需的关键字比较次数是该路径上内部节点的总数。

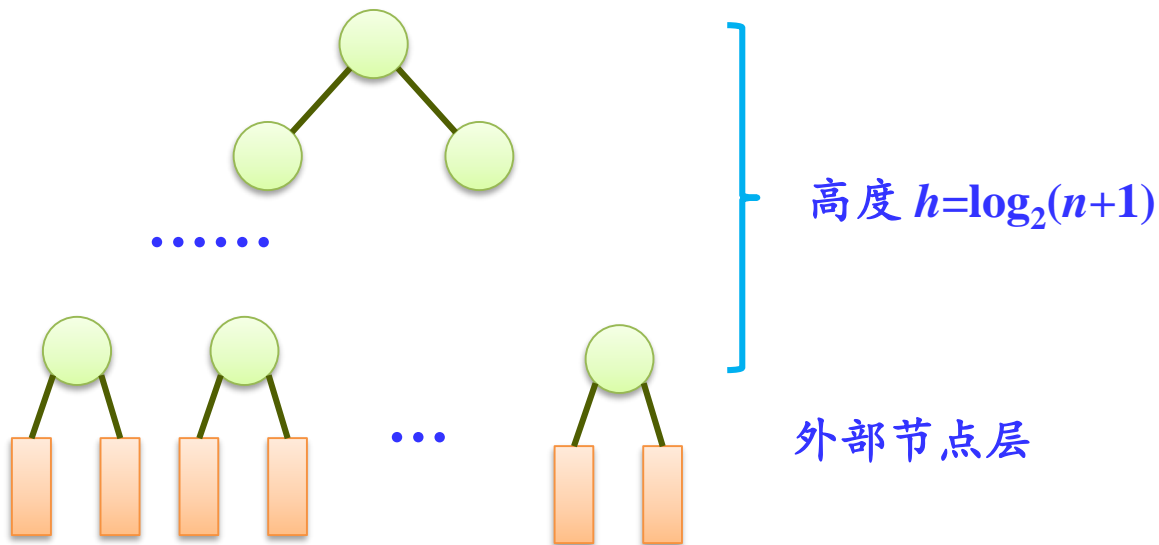


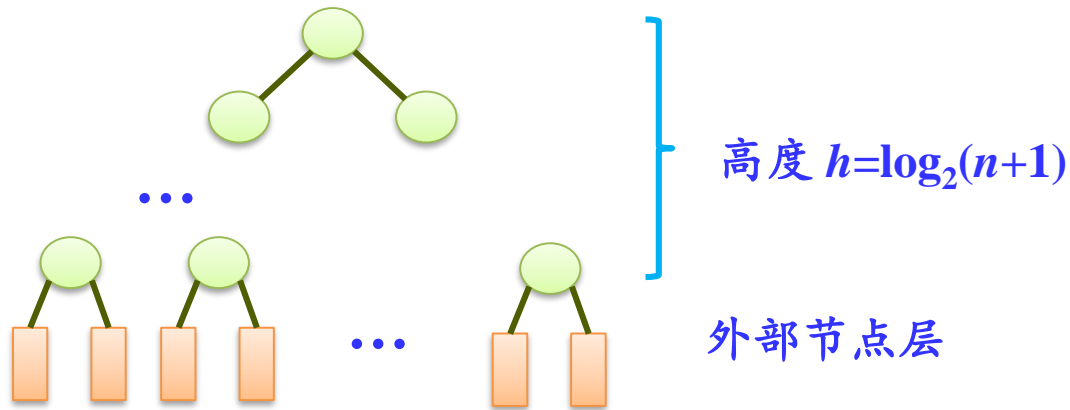
关键字比较3次

(3) 在查找不成功时，会找到图中某个外部节点，则不成功时的平均查找长度：

$$ASL_{\text{不成功}} = \frac{4 \times 3 + 8 \times 4}{12} = 3.67$$

当 n 比较大时，将判定树看成内部节点的总数为 $n=2^h-1$ 、高度为 $h=\log_2(n+1)$ 的**满二叉树**（高度 h 不计外部节点）。树中第 i 层上的记录个数为 2^{i-1} ，查找该层上的每个记录需要进行 i 次比较。





在等概率假设下，二分查找成功时的平均查找长度为：

$$ASL_{bn} = \sum_{i=1}^n p_i c_i = \frac{1}{n} \sum_{j=1}^h 2^{j-1} \times j = \frac{n+1}{n} \log_2(n+1) - 1 \approx \log_2(n+1) - 1$$

对于 n 个元素，二分查找，成功时最多的关键字比较次数为： $\lceil \log_2(n+1) \rceil$
不成功时关键字比较次数为： $\lceil \log_2(n+1) \rceil$ 。

二分查找的时间复杂度为 $O(\log_2 n)$ 。

数据结构经典算法的启示

顺序查找算法



利用了数据的有序性

二分查找算法

9.2.3 索引存储结构和分块查找

1、索引存储结构

索引存储结构 = 数据表 + 索引表

索引表中的每一项称为索引项，索引项的一般形式是：

(关键字，地址)

关键字唯一标识一个节点，地址作为指向该关键字对应节点的指针，也可以是相对地址。

示例

学号	姓名
1	张三
4	李四
3	王五
2	刘六

学生表



存储

数据表

存储地址

0
1
2
3

学号	姓名
1	张三
4	李四
3	王五
2	刘六

学号	地址
1	0
4	1
3	2
2	3

索引表

学号	地址
1	0
2	3
3	2
4	1

提取

排序

数据表

存储地址	学号	姓名
0	1	张三
1	4	李四
2	3	王五
3	2	刘六

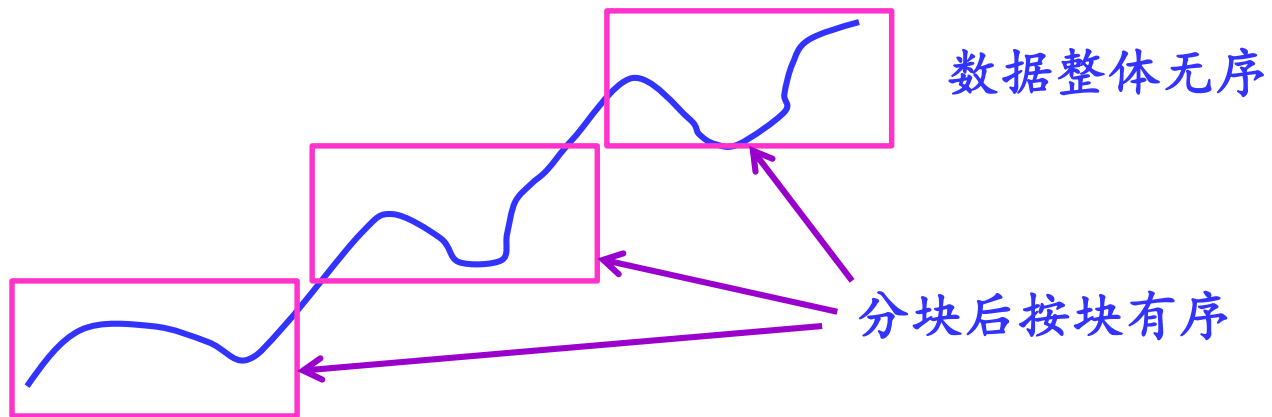
索引表

学号	地址
1	0
2	3
3	2
4	1

学生表的索引存储结构

2、分块查找

思路：

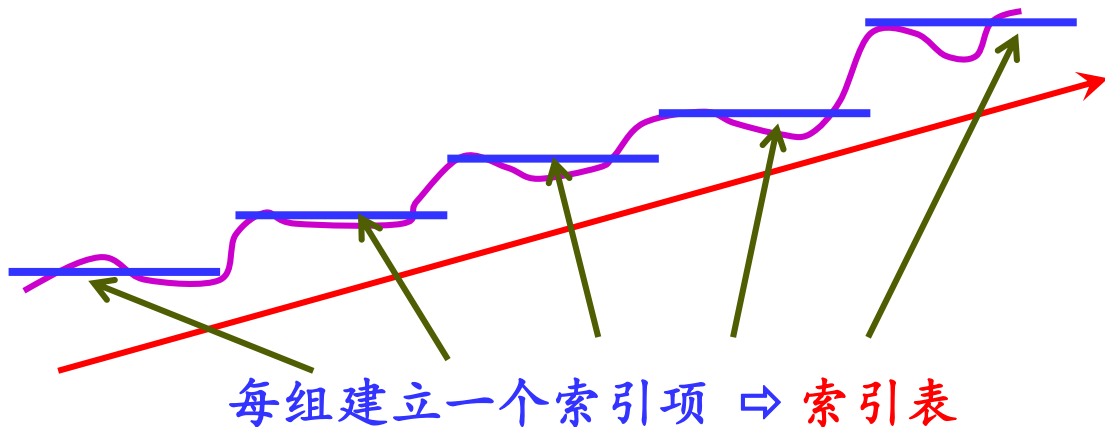


例如，设有一个线性表，其中包含25个记录，其关键字序列为：

8, 14, 6, 9, 10, 22, 34, 18, 19, 31, 40, 38, 54, 66, 46, 71, 78, 68, 80, 85, 100, 94, 88, 96, 87

分块：将 $n=25$ 个记录分为 $b=5$ 块，每块中有 $s=5$ 个记录。

数据特性：



分块查找方法：

- 索引表（有序）：可以顺序查找块，也可以二分查找块。
- 数据块（无序）：只能顺序查找块中元素。

分块查找演示

查找关键字为80的记录

索引表

14	34	66	85	100
0	5	10	15	20

key
link

8	14	6	9	10	22	34	18	19	31	40	38	54	66	46	71	78	68	80	85	100	94	88	96	87
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

数据表

分块查找的索引存储结构

- (1) 顺序查索引表，比较4次
- (2) 在对应块中查找，比较4次，共比较8次。

性能：

分块查找介于顺序查找和二分查找之间。



——本讲完——