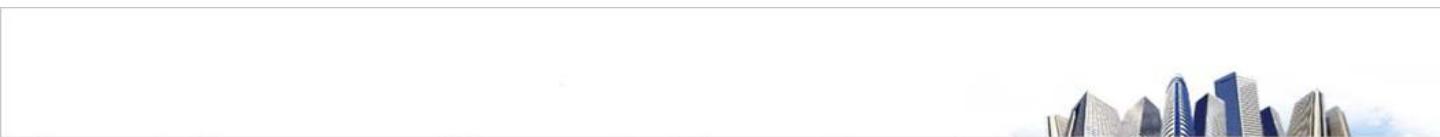




数据结构
(C语言版) (第2版)
排序
基数排序

主讲教师：汪红松



教学内容 Contents

- 
- 
- 1 排序的基本概念和方法
 - 2 插入排序
 - 3 交换排序
 - 4 选择排序
 - 5 归并排序
 - 6 基数排序

- ▶▶▶
- 一、基数排序**
 - 二、链式基数排序**
 - 三、排序算法比较**
 - 四、排序算法选择规则**



老师

▶▶▶ 一、基数排序

前面的排序方法主要通过关键字值之间的比较和移动而基数排序不需要关键字之间的比较。

对52张扑克牌按以下次序排序：

♣ 2 < ♣ 3 < < ♣ A < ♦ 2 < ♦ 3 < < ♦ A <

♥ 2 < ♥ 3 < < ♥ A < ♠ 2 < ♠ 3 < < ♠ A

两个关键字：花色 (♣ < ♦ < ♥ < ♠)

面值 (2 < 3 < < A)

并且 “花色” 地位高于 “面值”

▶▶▶ 一、基数排序

多关键字排序



最高位优先MSD (Most Significant Digit first)



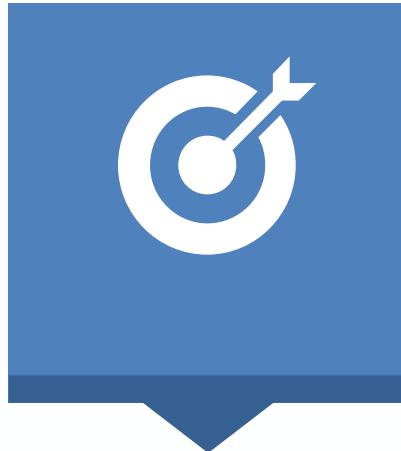
最低位优先LSD (Least Significant Digit first)

链式基数排序

链式基数排序：用链表作存储结构的基数排序。

▶▶▶ 一、基数排序

1.最高位优先法



先对**最高位**关键字 k_1 (如花色) 排序，将序列分成若干子序列，每个子序列有相同的 k_1 值；

然后让每个子序列对**次关键字** k_2 (如面值) 排序，又分成若干更小的子序列；

依次重复，直至就每个子序列对**最低位关键字** k_d 排序，就可以得到一个有序的序列。

✓ 十进制数比较可以看作是一个多关键字排序。

▶▶▶ 一、基数排序

1.最高位优先法

278,109,063,930,184,589,269,008,083

按百位排序

008,063,083 

按十位排序

008
063
083

109,184 

109
184

269,278 

269
278

589

589

930

930

▶▶▶ 一、基数排序

2. 最低位优先



首先依据**最低位**排序码 K_d 对所有对象进行一趟排序。



再依据**次低位**排序码 K_{d-1} 对上一趟排序结果排序。



依次重复，直到依据**排序码 K_1** 最后一趟排序完成，就可以得到一个有序的序列。

✓ 这种方法**不需要再分组**，而是整个对象组都参加排序。

▶▶▶ 一、基数排序

2.最低位优先法

278 , 109 , 063 , 930 , 184 , 589 , 269 , 008 , 083

按个位排序

930 , 063 , 083 , 184 , 278 , 008 , 109 , 589 , 269

按十位排序

008 , 109 , 930 , 063 , 169 , 278 , 083 , 184 , 589

按百位排序

008 , 063 , 083 , 109 , 169 , 184 , 278 , 589 , 930 ,

▶▶▶ 二、链式基数排序

– 先决条件：



知道各级关键字的**主次关系**



知道各级关键字的**取值范围**

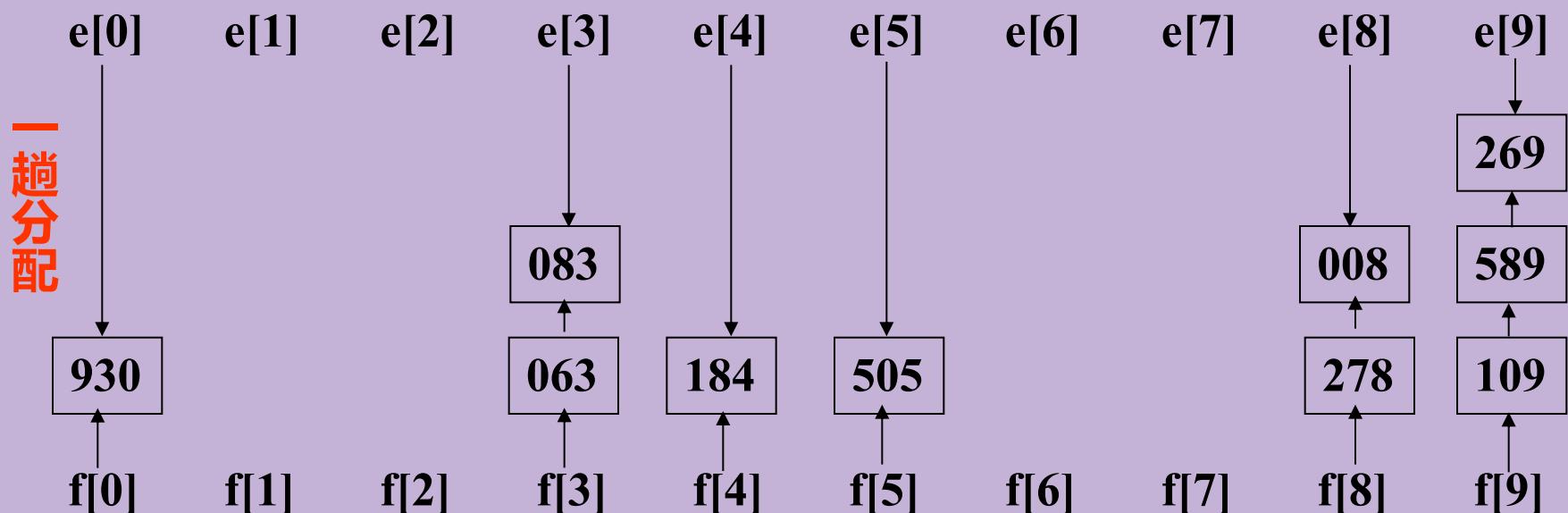
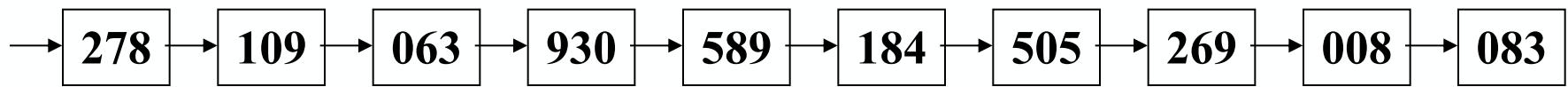
利用“**分配**”和“**收集**”对关键字进行排序。

– 过程

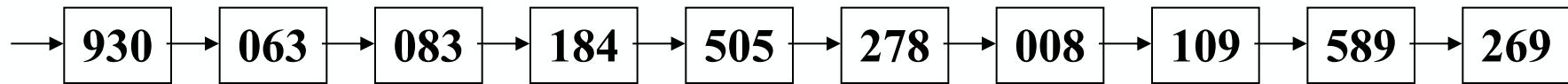
- 首先对**低位关键字**排序，各个记录按照此位关键字的值‘**分配**’到相应的序列里。
- 按照序列对应的值的大小，从各个序列中将记录‘**收集**’，收集后的序列按照此位关键字有序。
- 在此基础上，对前一位关键字进行排序。

二、链式基数排序

初始状态：

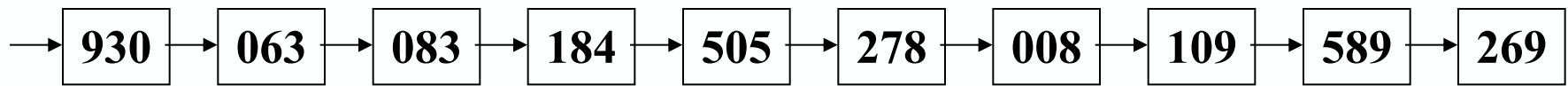


一趟收集：

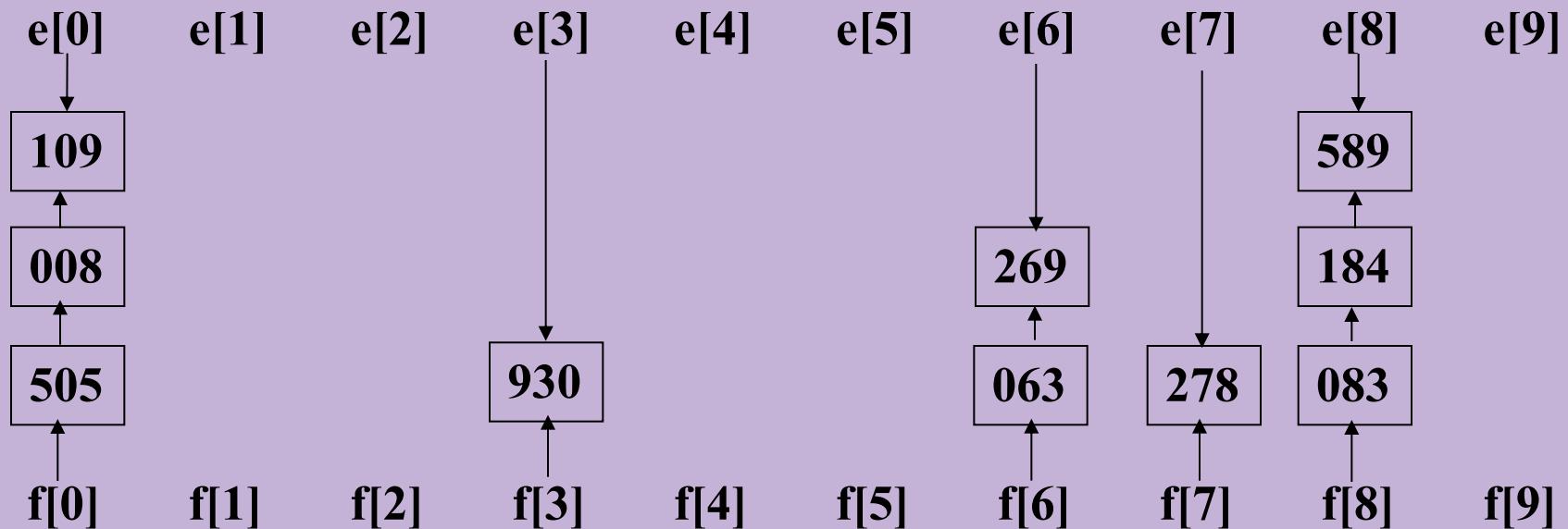


二、链式基数排序

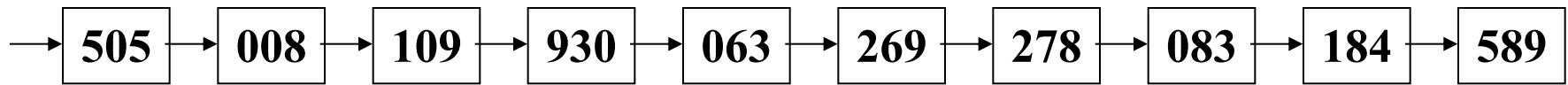
一趟收集



一趟分配

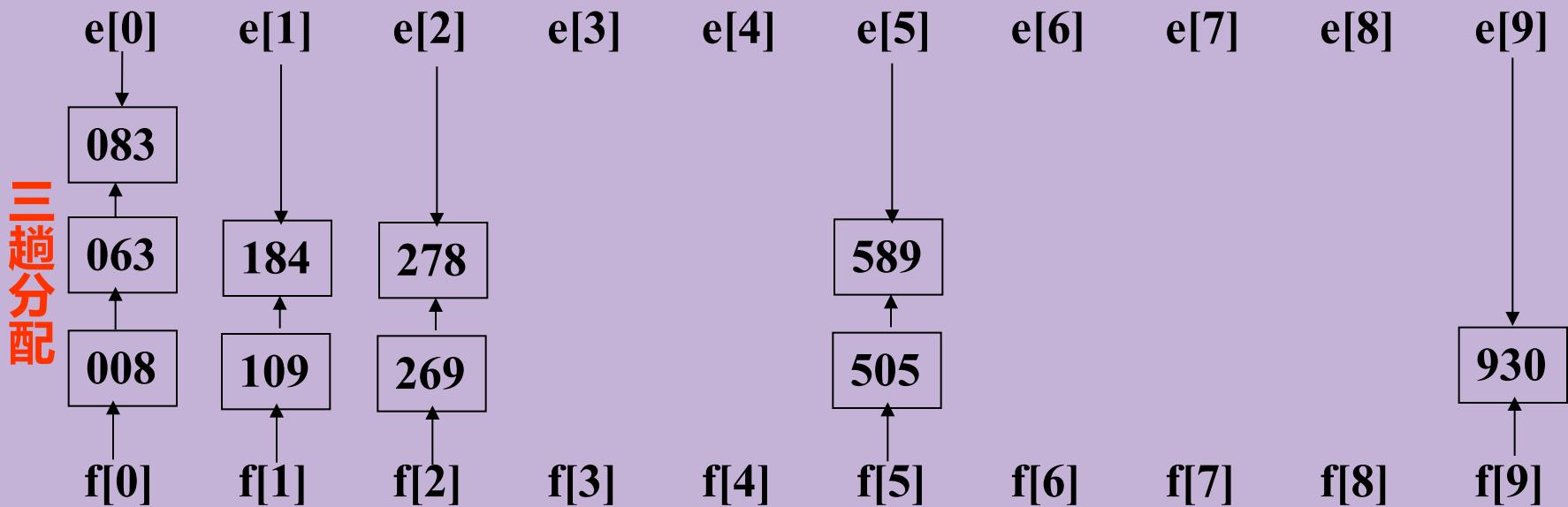
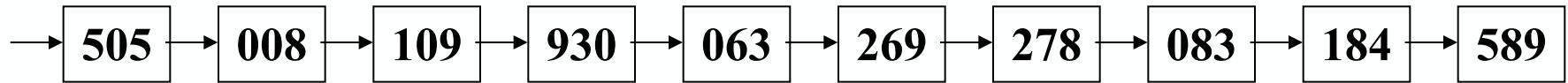


二趟收集：

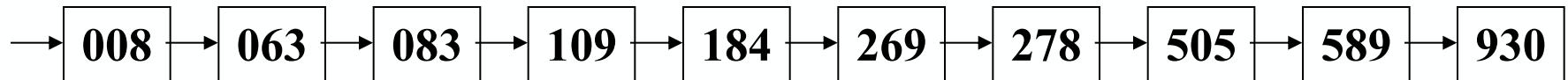


二、链式基数排序

二趟收集



三趟收集：



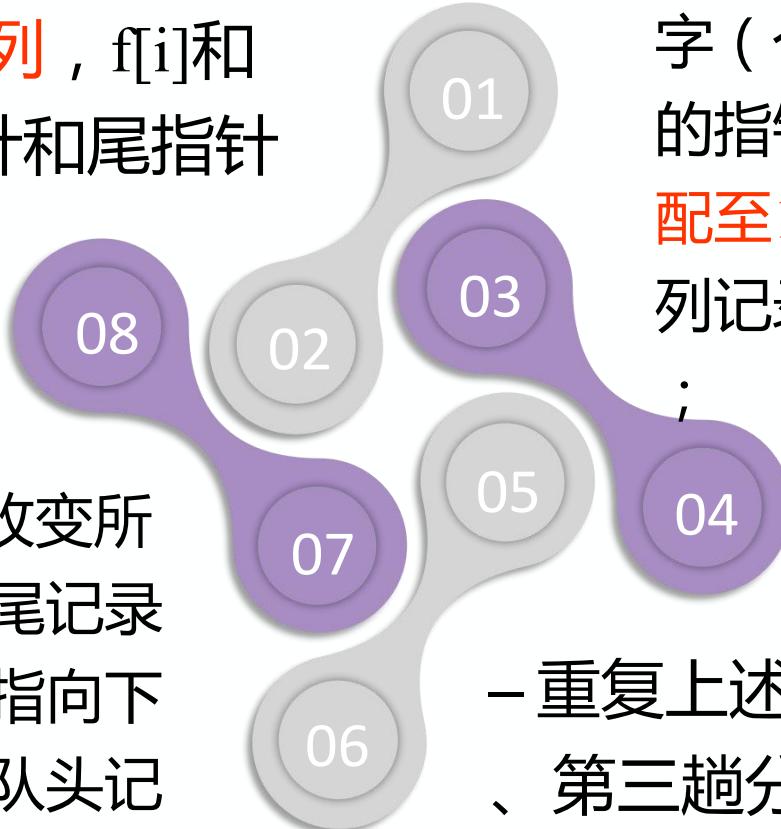
二、链式基数排序

1. 链式基数排序步

- 设置10个队列， $f[i]$ 和
 $e[i]$ 分别头指针和尾指针

；

- 第一趟收集是改变所
有非空队列的队尾记录
的指针域，令其指向下一
个非空队列的队头记
录，重新将10个队列链
成一个链表；



- 第一趟分配对最低位关键
字（个位）进行，改变记录
的指针值，将链表中记录分
配至10个链队列中，每个队
列记录的关键字的个位相同

；

- 重复上述两步，进行第二趟
、第三趟分配和收集，分别对
十位、百位进行，最后得到一
个有序序列。

二、链式基数排序

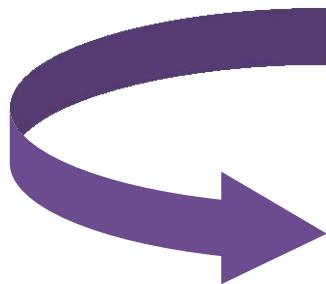
2. 算法分析

n个记录

每个记录有 d 位关键字

关键字取值范围rd(如十进制为10)

- 重复执行d趟 “分配” 与 “收集”
- 每趟对 n 个记录进行 “分配” , 对rd个队列进行 “收集”
- 需要增加n+2rd个附加链接指针。



- ✓ 时间效率 : $O(d(n+rd))$
- ✓ 空间效率 : $O(n+rd)$
- ✓ 稳定性 : 稳定

三、排序算法比较

排序方法	平均时间	比较次数		移动次数		稳定性	附加存储
		最好	最差	最好	最差		
直接插入	n^2	n	n^2	0	n^2	✓	1
折半插入	n^2	$n \log_2 n$		0	n^2	✓	1
希尔排序	$n^{1.3}$			0		✗	1
起泡排序	n^2	n	n^2	0	n^2	✓	1
快速排序	$n \log_2 n$	$n \log_2 n$	n^2	$n \log_2 n$	n^2	✗	$\log_2 n$
简单选择	n^2	n^2		0	n	✓	1
堆排序	$n \log_2 n$	$n \log_2 n$		$n \log_2 n$		✗	1
归并排序	$n \log_2 n$	$n \log_2 n$		$n \log_2 n$		✓	n
基数排序	$d(n + rd)$					✓	$n + rd$

(数据不是顺次后移时将导致方法不稳定)

三、排序算法比较



- 按平均时间排序方法分为四类
 $O(n^2)$ 、 $O(n \log n)$ 、 $O(n^{1+\varepsilon})$ 、 $O(n)$ ；



- 快速排序是基于比较的内部排序中平均性能最好的；



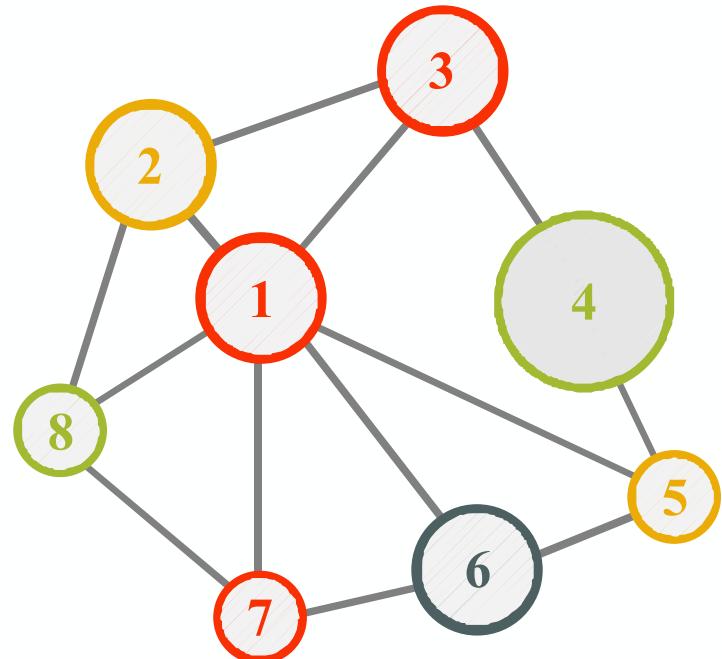
- 基数排序时间复杂度最低，但对关键字结构有要求
 - 。



知道各级关键字的主次关系
知道各级关键字的取值范围

▶▶▶ 三、排序算法比较

- ✓ 为避免顺序存储时大量移动记录的时间开销，可考虑用链表作为存储结构：直接插入排序、归并排序、基数排序；
- ✓ 不宜采用链表作为存储结构的折半插入排序、希尔排序、快速排序、堆排序。



▶▶▶ 四、排序算法选择规则

n较小时

- (1) 分布随机，稳定性不做要求，则采用**快速排序**；
- (2) 内存允许，要求排序稳定时，则采用**归并排序**；
- (3) 可能会出现正序或逆序，稳定性不做要求，则采用**堆排序**或**归并排序**。

n较大时

- (1) 基本有序，则采用**直接插入排序**；
- (2) 分布随机，则采用**简单选择排序**，若排序码不接近逆序，也可以采用直接插入排序。

▶▶▶ 小结

1. 多关键字的排序思想
2. 典型的分配类排序方法链式基数排序
3. 对排序算法进行了比较和归纳