
数据结构

排序

创客学院 小美老师

排序定义

稳定排序和非稳定排序

设文件 $f = (R_1 \dots R_i \dots R_j \dots R_n)$ 中记录 R_i 、 R_j ($i \neq j$, $i, j = 1 \dots n$) 的key相等, 即 $K_i = K_j$ 。

若在排序前 R_i 领先于 R_j , 排序后 R_i 仍领先于 R_j , 则称这种排序是稳定的, 其含义是它没有破坏原本已有序的次序。

排序定义

- **内排序和外排序**
- 若待排文件f在计算机的内存储器中，且排序过程也在内存中进行，称这种排序为内排序。
- 若排序中的文件存入外存储器，排序过程借助于内外存数据交换（或归并）来完成，则称这种排序为外排序。

内排序方法

各种内排序方法可归纳为以下五类：

(1) 插入排序

(2) 交换排序

(3) 选择排序

(4) 归并排序

.....

插入排序

直接插入排序

折半插入排序

链表插入排序

Shell（希尔）排序

.....

直接插入排序

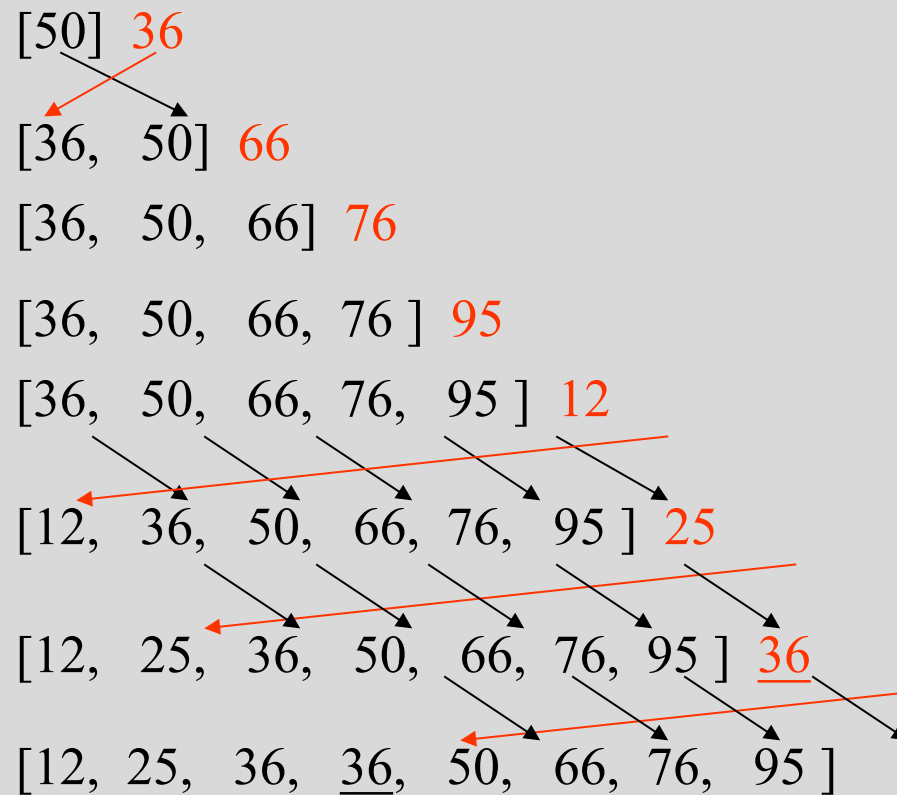
设待排文件 $f = (R_1 R_2 \dots R_n)$ 相应的key集合为 $k = \{k_1 k_2 \dots k_n\}$,

排序方法

先将文件中的 (R_1) 看成只含一个记录的有序子文件, 然后从 R_2 起, 逐个将 R_2 至 R_n 按key插入到当前有序子文件中, 最后得到一个有序的文件。插入的过程上是一个key的比较过程, 即每插入一个记录时, 将其key与当前有序子表中的key进行比较, 找到待插入记录的位置后, 将其插入即可。

直接插入排序

设文件记录的key集合 $k=\{50, 36, 66, 76, 95, 12, 25, \underline{36}\}$



折半插入排序

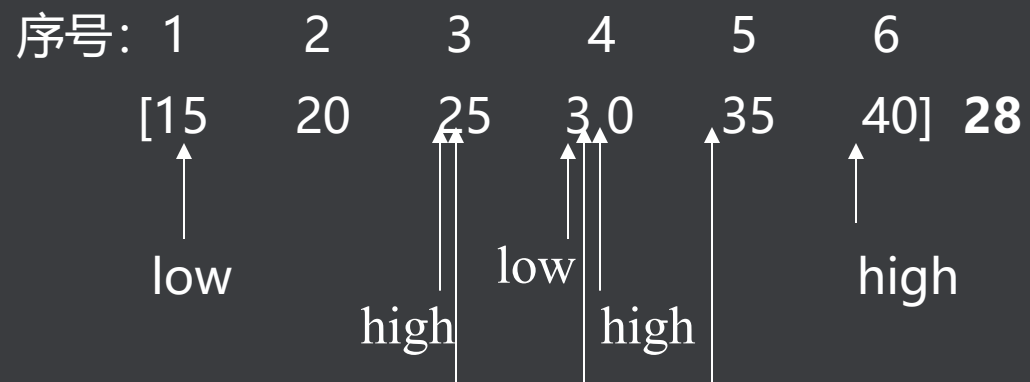
排序算法的 $T(n)=O(n^2)$,是内排序时耗最高的时间复杂度。

折半插入排序方法

先将 $(R[1])$ 看成一个子文件, 然后依次插入 $R[2].....R[n]$ 。但在插入 $R[i]$ 时, 子表 $[R[1].....R[i-1]]$ 已是有序的, 查找 $R[i]$ 在子表中的位置可按折半查找方法进行, 从而降低key的比较次数。

折半插入排序

设当前子表key序列及插入的 $k_i=28$ 如下：



令：

$$mid = \lfloor (low + high) / 2 \rfloor$$

链表插入排序

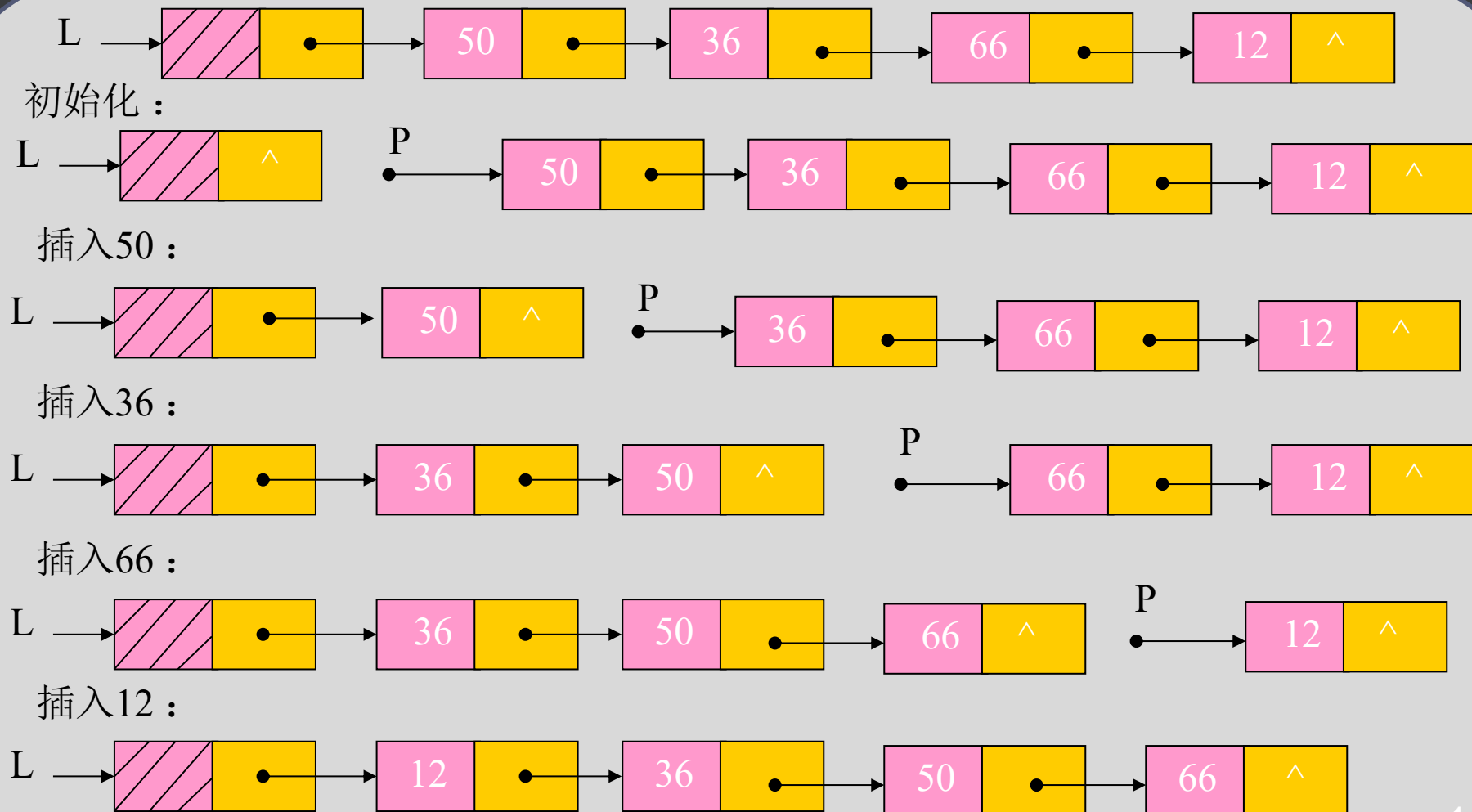
设待排序文件 $f = (R_1 R_2 \dots R_n)$ ，对应的存储结构为单链表结构



链表插入排序实际上是一个对链表遍历的过程。先将表置为空表，然后依次扫描链表中每个结点，设其指针为 p ，搜索到 p 结点在当前子表的适当位置，将其插入。

链表插入排序

设含4个记录的链表如图：



交换排序

“起泡” 排序 (Bubble Sort)

“快速” 排序 (Quick Sort)

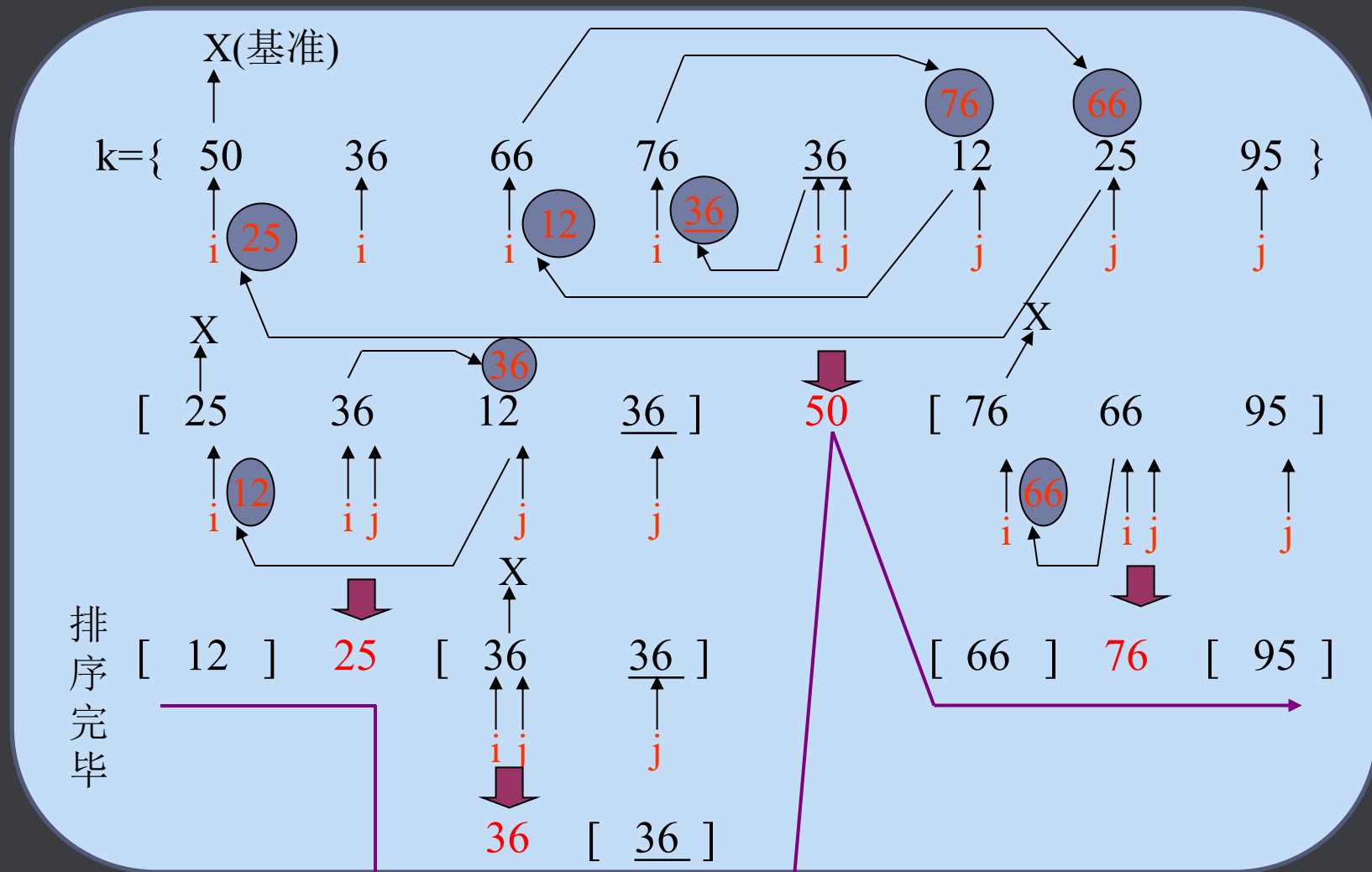
起泡排序

设记录key集合k={50, 36, 66, 76, 95, 12, 25, 36}, 排序过程如下:

K	第1趟	第2趟	第3趟	第4趟	第5趟	第6趟	
50	36	36	36	36	12	12	排序完毕
36	50	50	50	12	25	25	
66	66	66	12	25	36	36	
76	76	12	25	<u>36</u>	36		
95	12	25	<u>36</u>	50			
12	25	<u>36</u>	66				
25	<u>36</u>	76					
<u>36</u>	95						

快速排序

设记录的key集合 $k=\{50, 36, 66, 76, 36, 12, 25, 95\}$ ，每次以集合中第一个key为基准的快速排序过程如下：



扫一扫，获取更多信息



THANK YOU