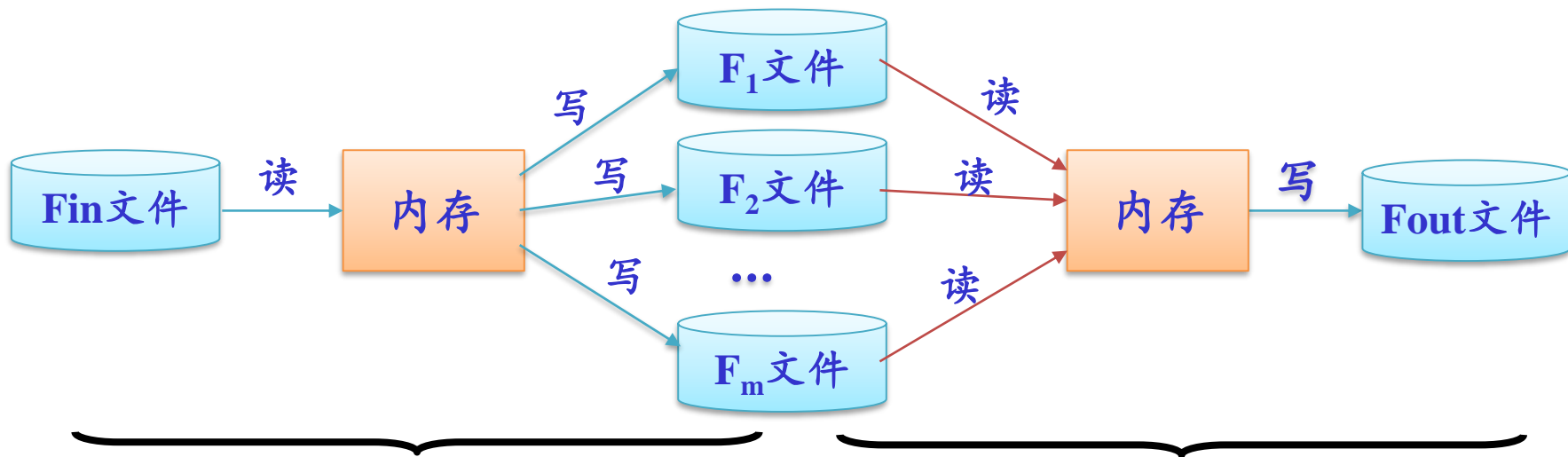


## 11.2 磁盘排序

### 磁盘排序过程



① 生成若干初始归并段

② 归并成一个有序文件

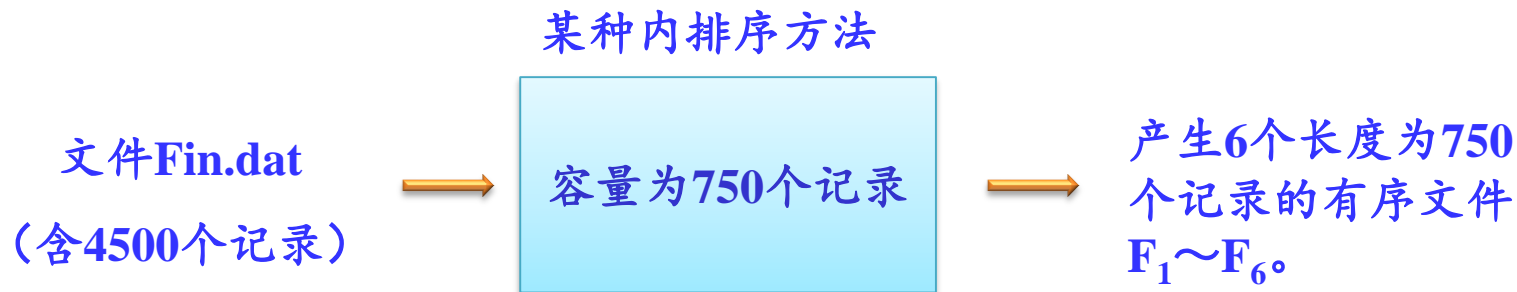
## 磁盘排序示例演示

设有一个文件Fin.dat，内含4500个记录： $A_1, A_2, \dots, A_{4500}$ ，现在要对该文件进行排序，结果放在Fout.dat文件中。可占用的内存空间至多只能对750个记录进行排序。

Fin.dat文件放在磁盘上。假设每个记录占用一个物理块。



## 第1阶段：产生初始归并段

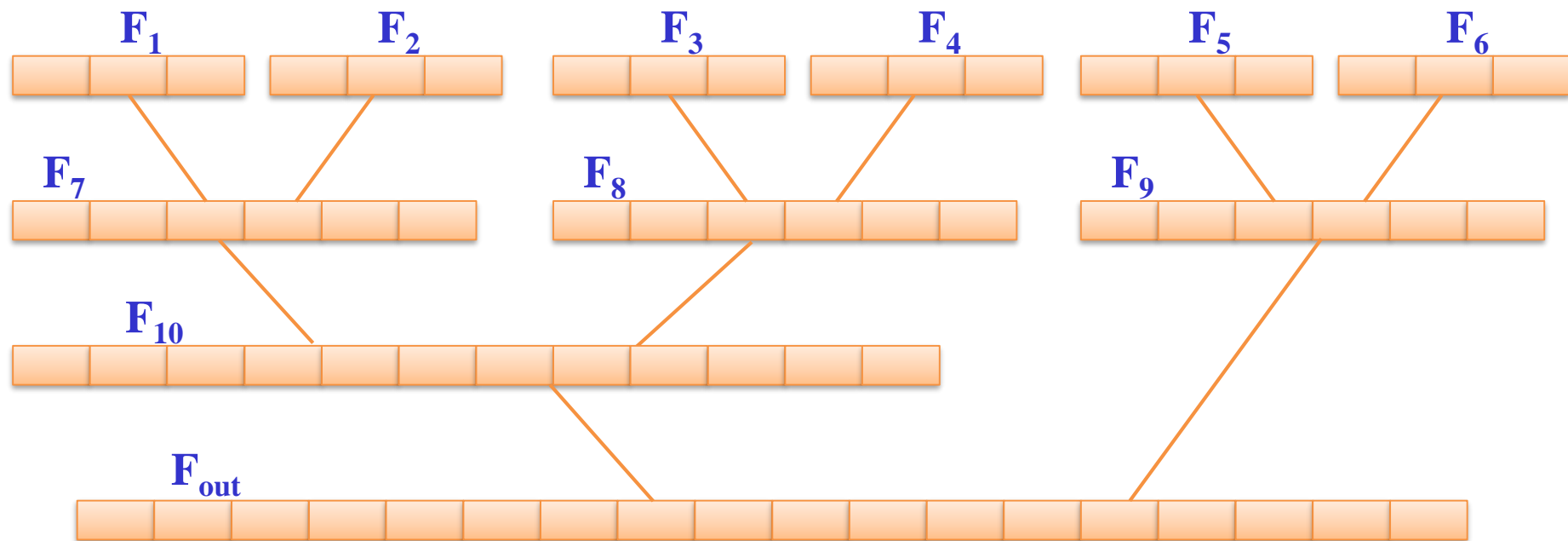


## 第2阶段：多路归并

可用内存空间大小为750个记录

可以采用多种归并方案来完成

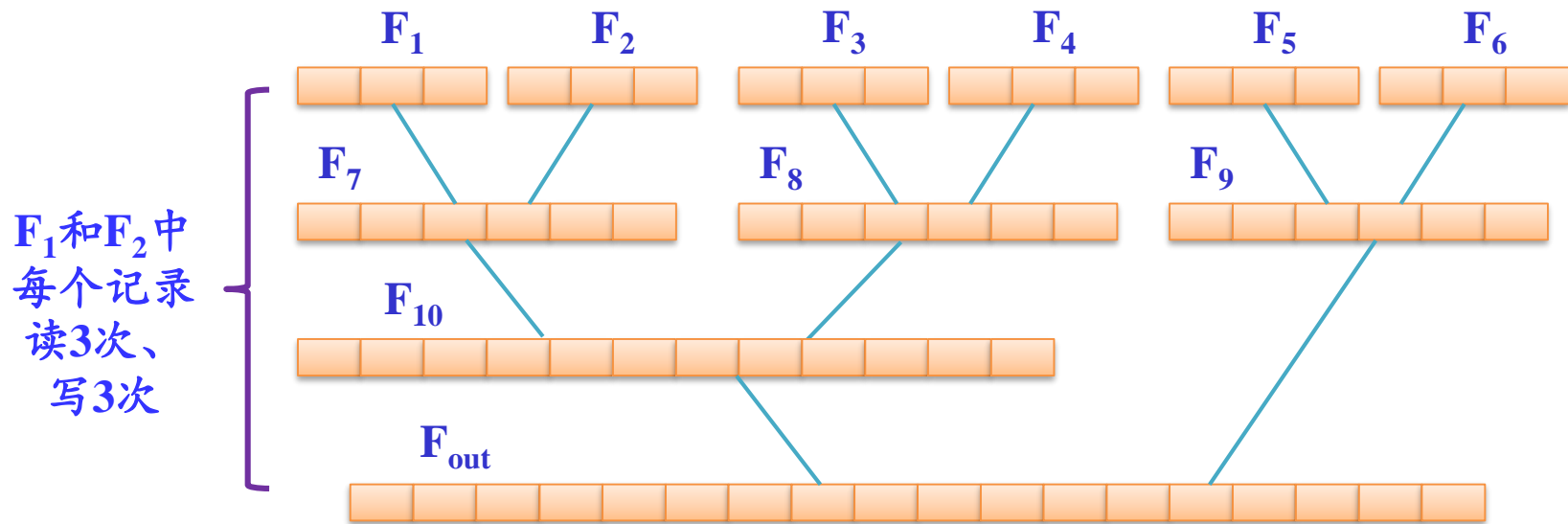
## 归并方案1：二路归并1



### 注意：

- ① 内存大小为750个记录，但任意大小的两个归并段都可以进行归并。
- ② 每归并一次，参与归并的每个记录都要读一次和写一次。

## 方案1的读写记录数计算

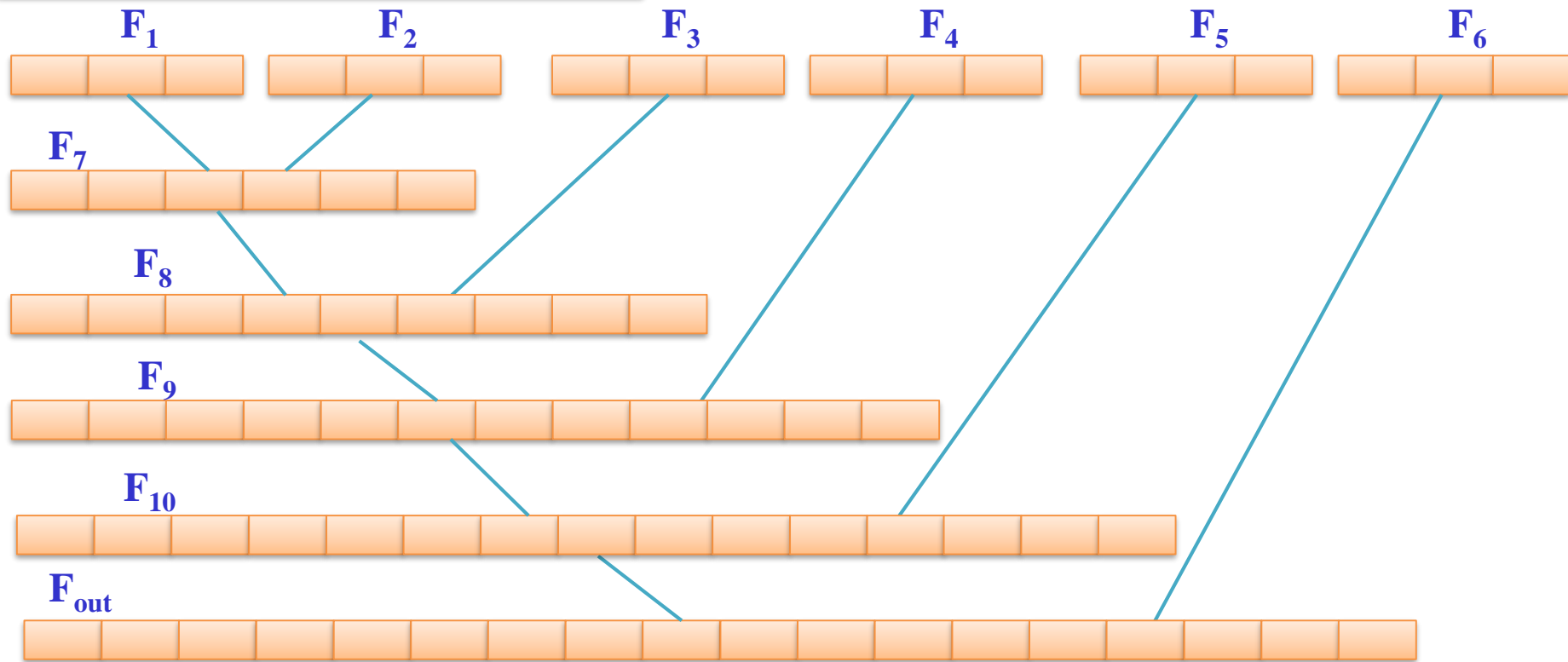


总的读记录数（写记录数与之相同）：

$$[(F_1+F_2+F_3+F_4 \text{ 的记录数}) \times 3 + (F_5+F_6 \text{ 的记录数}) \times 2] = 12000 = 8/3 \text{ 遍}$$

该数越大，效率越差      等于哈夫曼树的WPL

## 归并方案2：二路归并2

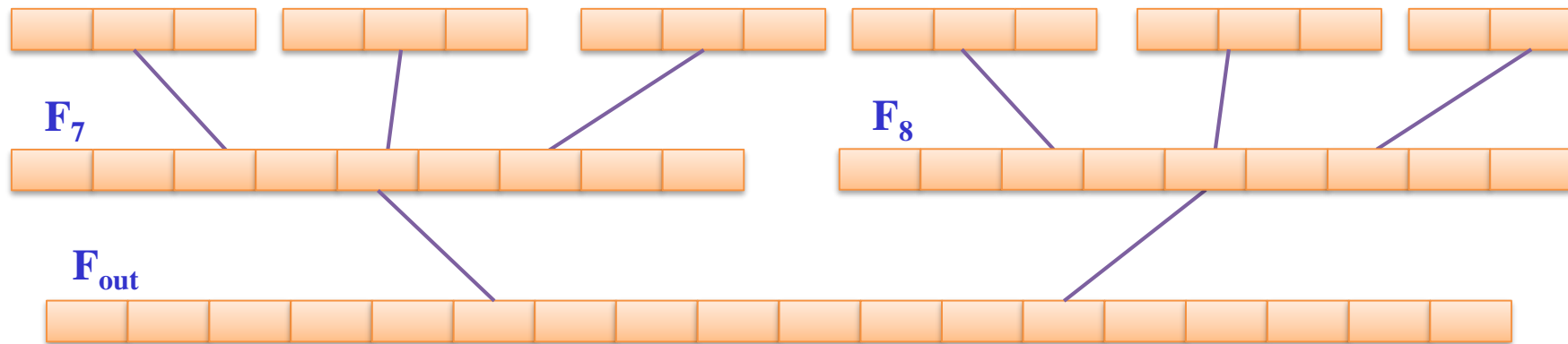


方案2：总的读记录数  $WPL=15000$ 。

方案1：总的读记录数  $WPL=12000$ 。

方案1更好

### 归并方案3：三路归并



总的读记录数（写记录数与之相同）：

$$\text{方案3: } WPL = (750 + 750 + 750) \times 2 + (750 + 750 + 750) \times 2 = 9000$$

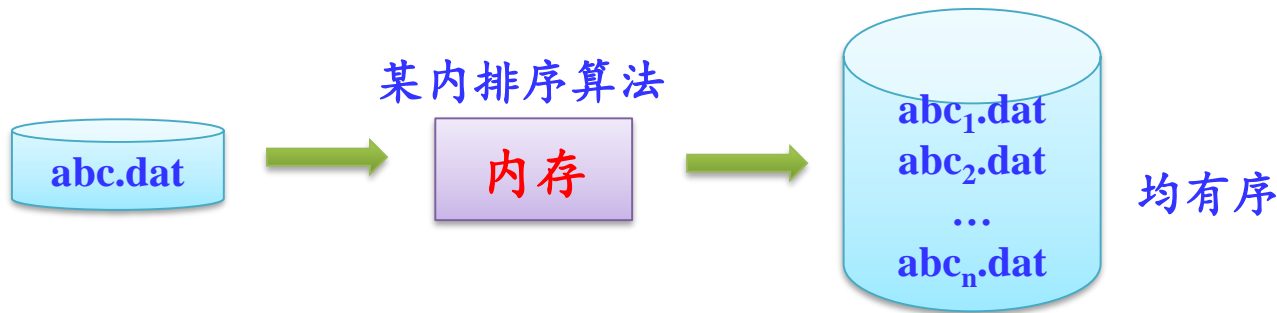


不同的归并方案所需要的读写记录数是不同的！



## 11.2.1 生成初始归并段

生成前面的初始归并段的方法



另一种方法：采用一种称为置换-选择排序方法用于生成初始归并段。



可以减少生成的初始归并段个数

## 置换—选择排序方法

- (1) 从待排文件 $F_{in}$ 中按内存工作区WA的容量 $w$ 读入 $w$ 个记录。设归并段编号 $i=1$ 。
- (2) 使用败者树从WA中选出关键字最小的记录 $R_{min}$ 。
- (3) 将 $R_{min}$ 记录输出到 $F_{out}$ 中，作为当前归并段的一个成员。
- (4) 若 $F_{in}$ 不空，则从 $F_{in}$ 中读入下一个记录 $x$ 放在 $R_{min}$ 所在的工作区位置代替 $R_{min}$ 。
- (5) 在工作区中所有 $\geq R_{min}$ 的记录中选择出最小记录作为新的 $R_{min}$ ，转(3)，直到选不出这样的 $R_{min}$ 。
- (6) 设 $i=i+1$ ，开始一个新的归并段。
- (7) 若工作区已空，则初始归并段已全部产生；否则转(2)。

**【例11-1】** 设磁盘文件中共有18个记录，记录的关键字分别为：

{15,4,97,64,17,32,108,44,76,9,39,82,56,31,80,73,255,68}

若内存工作区可容纳5个记录，用置换-选择排序可产生几个初始归并段，每个初始归并段包含哪些记录？

## 置换-选择排序示例演示

18个记录 ( $w=5$ ) :

15 4 97 64 17 32 108 44 76 9 39 82 56 31 80 73 255 68  $\infty$



内存工作区  $w=5$

$R_{\min} = 9$

归并段1:

归并段2:

依次类推, 产生归并段2: 9, 31, 39, 56, 68, 73, 80, 255

## 置换-选择排序中关键字比较次数分析

共有 $n$ 个记录，内存工作区WA的容量为 $w$ ：

- 若在 $w$ 个记录中选取最小关键字的采用简单比较方法，每次需要 $w-1$ 次比较。
- 总的时间复杂度为 $O(nw)$ 。

——本讲完——