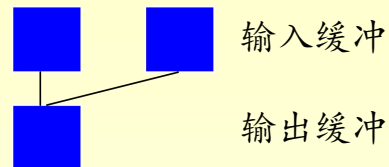
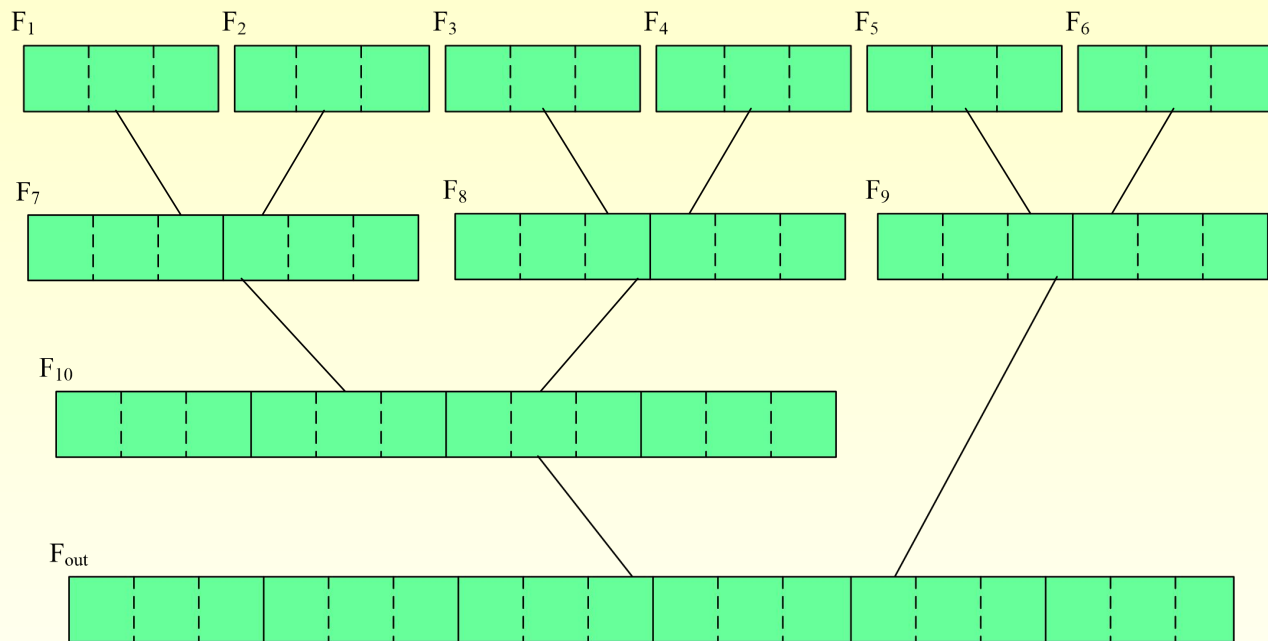




本节主题:

初始归并段的置换-选择排序算法

k路平衡归并



为什么要平衡?

为什么不可以产生尽可能长的初始归并段?

初始归并段的置换-选择排序算法

问题(示例)

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}

内存工作区可容纳 5 个记录

用置换-选择排序产生归并段的步骤

生成初始归并段 - {15,4,97,64,17,32,108,44,76,9,39,82,56,31,80,73,255,68}

(1) 读入与内存块大小相当的记录

(2) 从块中选择最小的记录Rmin

(3) 将Rmin加入当前归并段

读入记录	内存工作区状态	Rmin	输出之后的初始归并段状态
15,4,97,64,17	15,4,97,64,17	4(i=1)	1:{4}
32	15,32,97,64,17	15(i=1)	1:{4,15}
108	108,32,97,64,17	17(i=1)	1:{4,15,17}
44	108,32,97,64,44	32(i=1)	1:{4,15,17,32}
76	108,76,97,64,44	44(i=1)	1:{4,15,17,32,44}
9	108,76,97,64,9	64(i=1)	1:{4,15,17,32,44,64}
39	108,76,97,39,9	76(i=1)	1:{4,15,17,32,44,64,76}
82	108,82,97,39,9	82(i=1)	1:{4,15,17,32,44,64,76,82}
56	108,56,97,39,9	97(i=1)	1:{4,15,17,32,44,64,76,82,97}
31	108,56,31,39,9	108(i=1)	1:{4,15,17,32,44,64,76,82,97,108}
80	80,56,31,39,9	9(i=2)	2:{9}
73	80,56,31,39,73	31(i=2)	2:{9,31}
255	80,56,255,39,73	39(i=2)	2:{9,31,39}
68	80,56,255,68,73	56(i=2)	2:{9,31,39,56}
	80,,255,68,73	68(i=2)	2:{9,31,39,56,68}
	80,,255,,73	73(i=2)	2:{9,31,39,56,68,73}
	80,,255,,	80(i=2)	2:{9,31,39,56,68,73,80}
	,,255,,	255(i=2)	2:{9,31,39,56,68,73,80,255}

(4) 读入下一个记录, 替换刚输出的记录

(5) 从内存块中, 在大于等于Rmin的记录中, 找到最小的记录, 作为新的Rmin

(7) 若Fin空, 不再读入

(6) 若Rmin不存在, 开始一个新的归并段

(8) 若内存块已经空, 则结束

置换-选择排序算法步骤

- (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) 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) 在工作区中所有大于或等于 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)。

算法评价

- ❏ 用置换-选择排序算法的目标是，生成尽可能长的初始归并段。
- ❏ 决定初始归并段长度的因素
 - ❏ 内存工作区的大小
 - ❏ 输入文件中记录本来的排列次序
- ❏ 输入文件中的记录按其关键字随机排列时，所得到的初始归并段的平均长度，为内存工作区大小的2倍。

