



# 两路归并

# 外排序

大多数内排序算法都是利用了内存是直接访问的事实,读写一个数据是常量的时间。如果输入是在磁带上,磁带上的元素只能顺序访问。甚至数据是在磁盘上,效率还是下降,因为转动磁盘和移动磁头会产生延迟。

- ④ 外排序模型

- ④ 预处理

- ④ 归并

# 归 并

- ◎ 两路归并
- ◎ 多路归并
- ◎ 多阶段归并

# 两路归并

- ④ 假设我们有四条磁带A1,A2,B1和B2，两个用于输入，两个用于输出。开始时数据在A1上
- ④ 内存一次能排序M个记录
- ④ 工作过程：
  1. 从输入磁带上一次读入M个记录，对它们进行内排序，然后把已排序片段**轮流**写到B1和B2。回绕所有的磁带。--**预处理**
  2. 取每条磁带上的第一个已排序片段，把它们归并起来，并把结果写到A1。然后，从每条磁带上取下一个已排序片段，把它们归并起来，结果写到A2。继续这个过程，**轮流**把结果写到A1和A2，
  3. 回绕四条磁带，重复同样的步骤，这次使用A磁带作为输入，而B磁带作为输出。
  4. 重复步骤二和三，**直到剩下一个长度为N的已排序片断**

- 初始的磁带配置

A1	81	94	11	96	12	35	17	99	28	58	41	75	15
A2													
B1													
B2													

- 长度为3的已排序片段在两条磁带上的分布

A1													
A2													
B1	11	81	94	17	28	99	15						
B2	12	35	96	41	58	75							

- |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| A1 |    |    |    |    |    |    |    |
| A2 |    |    |    |    |    |    |    |
| B1 | 11 | 81 | 94 | 17 | 28 | 99 | 15 |
| B2 | 12 | 35 | 96 | 41 | 58 | 75 |    |

- |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| A1 | 11 | 12 | 35 | 81 | 94 | 96 | 15 |
| A2 | 17 | 28 | 41 | 58 | 75 | 99 |    |
| B1 |    |    |    |    |    |    |    |
| B2 |    |    |    |    |    |    |    |

- |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| A1 | 11 | 12 | 35 | 81 | 94 | 96 | 15 |
| A2 | 17 | 28 | 41 | 58 | 75 | 99 |    |
| B1 |    |    |    |    |    |    |    |
| B2 |    |    |    |    |    |    |    |

- [illegible]

- 经过第三轮归并后的磁带

A1	11	12	15	17	28	35	41	58	75	81	94	96	99
A2													
B1													
B2													





## 时间分析

- ④ 算法需要  $\lceil \log(N/M) \rceil$  次的归并处理，加上初始已排序片段的构造处理。
- ④ 如前例排序13个数据，每个已排序片断长度为3。因此归并次数为3。