

选第二大

选第二大

输入： n 个数的数组 L

输出： 第二大的数 $second$

通常算法： 顺序比较

1. 顺序比较找到最大 max
2. 从剩下 $n - 1$ 个数中找最大，就是第二大 $second$

时间复杂度：

$$W(n) = n - 1 + n - 2 = 2n - 3$$

提高效率的途径

- 成为第二大数的条件：仅在与最大数的比较中被淘汰.
- 要确定第二大数，必须知道最大数.
- 在确定最大数的过程中记录下被最大数直接淘汰的元素.
- 在上述范围（被最大数直接淘汰的数）内的最大数就是第二大数.
- 设计思想： 用空间换时间.

锦标赛算法

1. 两两分组比较，大者进入下一轮，直到剩下 1 个元素 *max* 为止
2. 在每次比较中淘汰较小元素，将被淘汰元素记录在淘汰它的元素的链表上
3. 检查 *max* 的链表，从中找到最大元，即 *second*

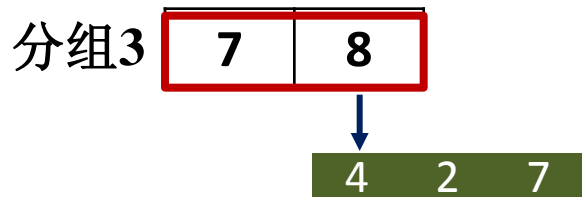
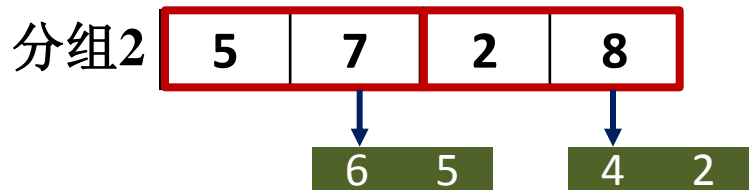
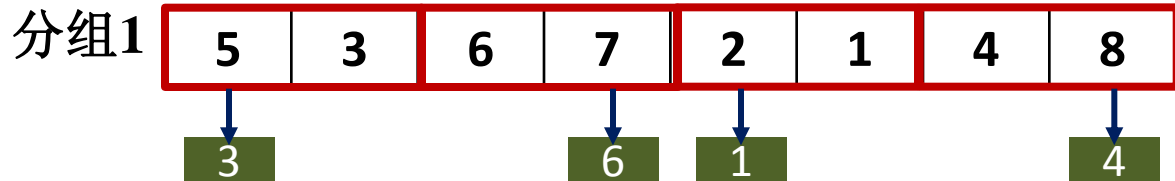
伪码

算法 FindSecond

输入: n 个数的数组 L , 输出: $second$

1. $k \leftarrow n$ // 参与淘汰的元素数
 2. 将 k 个元素两两1组, 分成 $\lfloor k/2 \rfloor$ 组
 3. 每组的2个数比较, 找到较大数
 4. 将被淘汰数记入较大数的链表
 5. if k 为奇数 then $k \leftarrow \lfloor k/2 \rfloor + 1$
 6. else $k \leftarrow \lfloor k/2 \rfloor$
 7. if $k > 1$ then goto 2
 8. $max \leftarrow$ 最大数
 9. $second \leftarrow max$ 的链表中的最大
- 一轮淘汰
- 继续分组淘汰

实例



时间复杂度分析

命题1 设参与比较的有 t 个元素，经过 i 轮淘汰后元素数至多为 $\lceil t/2^i \rceil$.

证 对 i 归纳. $i=1$, 分 $\lfloor t/2 \rfloor$ 组，淘汰 $\lfloor t/2 \rfloor$ 个元素，进入下一轮元素数是 $t - \lfloor t/2 \rfloor = \lceil t/2 \rceil$

假设 i 轮分组淘汰后元素数至多为 $\lceil t/2^i \rceil$ ，那么 $i+1$ 轮分组淘汰后元素数为

$$\lceil \lceil t/2^i \rceil / 2 \rceil = \lceil t/2^{i+1} \rceil$$

时间复杂度分析（续）

命题2 max 在第一阶段分组比较中总计进行了 $\lceil \log n \rceil$ 次比较.

证 假设到产生 max 时总计进行 k 轮淘汰, 根据命题 1 有 $\lceil n/2^k \rceil = 1$.

若 $n=2^d$, 那么有

$$k = d = \log n = \lceil \log n \rceil$$

若 $2^d < n < 2^{d+1}$, 那么

$$k = d + 1 = \lceil \log n \rceil$$

时间复杂度分析（续）

第一阶段元素数： n

比较次数： $n-1$

淘汰了 $n-1$ 个元素

第二阶段：元素数 $\lceil \log n \rceil$

比较次数： $\lceil \log n \rceil - 1$

淘汰元素数为 $\lceil \log n \rceil - 1$

时间复杂度是

$$\begin{aligned} W(n) &= n - 1 + \lceil \log n \rceil - 1 \\ &= n + \lceil \log n \rceil - 2. \end{aligned}$$

小结

求第二大算法

- 调用2次找最大: $2n-3$
 - 锦标赛算法: $n + \lceil \log n \rceil - 2$
- 主要的技术: 用空间换时间