选最大及最小问题的时间复杂度分析

选择算法的时间复杂度分析

下界证明方法：构造最坏输入

- 任意给定一个算法 A，A对于任意输入 x 都存在一个确定的操作序列 τ
- τ中的操作分成两类：
  - 决定性的：能够对确定输出结果提供有效信息
  - 非决定性的：对确定结果没有帮助的冗余操作
- 根据算法A构造某个输入实例 x，使得A对x 的操作序列 τ 包含尽量多的非决定性操作.
- 给出冗余操作+必要的操作的计数公式

选择算法的有关结果

	算法	最坏情况	空间
选最大	顺序比较	$n-1$	$O(1)$
选最大和最小	顺序比较	$2n-3$	$O(1)$
	算法 FindMaxMin	$\lceil 3n/2 \rceil-2$	$O(1)$
选第二大	顺序比较	$2n-3$	$O(1)$
	锦标赛方法	$n+\lceil \log n \rceil-2$	$O(n)$
选中位数	排序后选择	$O(n\log n)$	$O(\log n)$
	算法Select	$O(n)$~$2.95n$	$O(\log n)$

选最大算法 Findmax是最优的算法

选最大与最小算法

定理6 任何通过比较找最大和最小的算法至少需要$\lceil 3n/2 \rceil$-2次比较.

证明思路：任给算法*A*，根据算法 *A* 的比较结果构造输入 *T*, 使得 *A* 对 *T* 至少做$\lceil 3n/2 \rceil$-2 次比较.

证：不妨设*n*个数彼此不等，*A*为任意找最大和最小的算法. max是最大，*A*必须确定有*n*-1个数比max小，通过与max的比较被淘汰. min是最小，*A*也必须确定有*n*-1个数比min大，通过与min的比较而淘汰.总共需要2*n*-2个信息单位.

基本运算与信息单位

数的状态标记及其含义：

N：没有参加过比较　　W：赢

L：输　　WL：赢过且至少输1次

如果比较后数的状态改变，则提供信息单位，状态不变不提供信息单位，每增加 1 个W 提供 1个信息单位每增加 1 个L 提供 1 个信息单位.

两个变量通过一次比较增加的信息单位个数不同：0,1,2

case1： N,N → W,L：增加2个信息单位

case2： W,N → W,L：增加1个信息单位

case3： W,L → W,L：增加0个信息单位

算法输出与信息单位

算法输出的条件：

n-2 个数带有 W 和 L 标记，最大数只带 W 标记，最小数只带 L 标记，总计 2n-2个信息单位

对于任意给定的算法，构造输入的原则是：

根据算法的比较次序，针对每一步参与比较的两个变量的状态，调整对参与比较的两个变量的赋值，使得每次比较后得到的信息单位数达到最小. 从而使得为得到 输出所需要的 2n-2个信息单位，该算法对所构造的输入至少要做 $\lceil 3n/2 \rceil$-2 次比较.

对输入变量的赋值原则

x 与 y 的状态	赋值策略	新状态	信息单位
N,N	$x>y$	**W,L**	**2**
W,N；WL,N	$x>y$	**W,L；WL,L**	**1**
L,N	$x<y$	**L,W**	**1**
W,W	$x>y$	**W,WL**	**1**
L,L	$x>y$	**WL,L**	**1**
W,L；WL,L；W,WL	$x>y$	不变	**0**
WL,WL	保持原值	不变	**0**

一个赋值的实例

x_1,x_2---$x_1>x_2$; x_1,x_5---$x_1>x_5$; x_3,x_4---$x_3>x_4$; x_3,x_6---$x_3>x_6$

x_3,x_1---$x_3>x_1$; x_2,x_4---$x_2>x_4$; x_5,x_6---$x_5>x_6$; x_6,x_4---$x_6>x_4$ …

	x_1		x_2		x_3		x_4		x_5		x_6	
	状态	值	状态	值	状态	值	状态	值	状态	值	状态	值
	N	*	N	*	N	*	N	*	N	*	N	*
$x_1>x_2$	W	20	L	10								
$x_1>x_5$	W	20							L	5		
$x_3>x_4$					W	15	L	8				
$x_3>x_6$					W	15					L	12
$x_3>x_1$	WL	20			W	25						
$x_2>x_4$			WL	10			L	8				
$x_5>x_6$									WL	5	L	3
$x_6>x_4$							L	2			WL	3

构造的输入为（20，10，25，2，5，3）

问题复杂度的下界

为得到2*n*-2个信息单位，对上述输入*A*至少做$\lceil 3n/2\rceil - 2$ 次比较.

一次比较得到2个信息单位只有case1. *A*至多有$\lfloor n/2\rfloor$个case1，至多得到 $2\lfloor n/2\rfloor \le n$个信息单位. 其它case, 1次比较至多获得1个信息单位，至少还需要 *n*-2次比较.

当 *n* 为偶数，*A*做的比较次数至少为

$$\lfloor n/2\rfloor + n-2 = 3n/2 - 2 = \lceil 3n/2\rceil - 2$$

当 *n* 为奇数，*A*做的比较次数至少为

$$\lfloor n/2\rfloor + n-2 + 1 = (n-1)/2 + 1 + n-2 = \lceil 3n/2\rceil - 2$$

结论：FindMaxMin是最优算法