

选最大与最小

选择问题

输入：集合 L (含 n 个不等的实数)

输出： L 中第 i 小元素

$i=1$, 称为最小元素

$i=n$, 称为最大元素

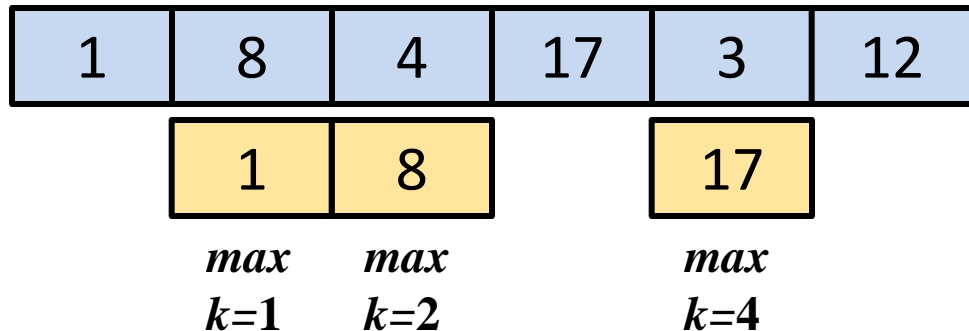
位置处在中间的元素，称为中位元素

n 为奇数，中位数唯一， $i = (n+1)/2$

n 为偶数，可指定 $i = n/2+1$

选最大

算法：顺序比较



输出： $max = 17$, $k=4$

算法最坏情况下的时间 $W(n)=n-1$

伪码

算法 Findmax

输入： n 个数的数组 L

输出： max, k

1. $max \leftarrow L[1]$

2. for $i \leftarrow 2$ to n do

3. if $max < L[i]$

4. then $max \leftarrow L[i]$

5. $k \leftarrow i$

6. return max, k

选最大最小

通常算法:

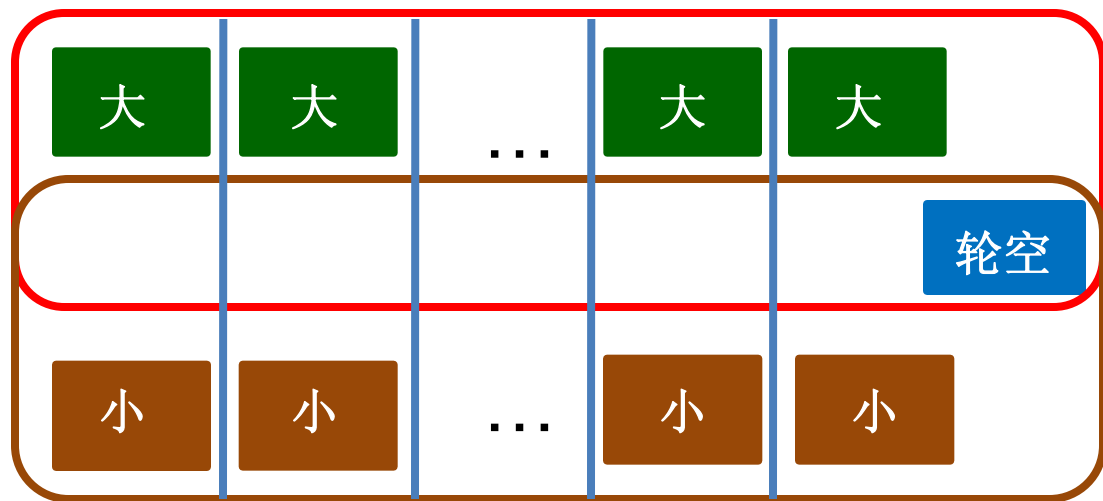
1. 顺序比较, 先选最大 max
2. 顺序比较, 在剩余数组中选最小 min , 类似于选最大算法, 但比较时保留较小的数

时间复杂性:

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

分组算法

组1 组2 ... 组 $\lfloor n/2 \rfloor$



伪码

算法 FindMaxMin

输入： n 个数的数组 L

输出： max , min

1. 将 n 个元素两两一组分成 $\lfloor n/2 \rfloor$ 组
2. 每组比较，得到 $\lfloor n/2 \rfloor$ 个较小和 $\lfloor n/2 \rfloor$ 个较大
3. 在 $\lceil n/2 \rceil$ 个较大（含轮空元素）中找最大 max
4. 在 $\lceil n/2 \rceil$ 个较小（含轮空元素）中找最小 min

最坏情况时间复杂度

行2 的组内比较: $\lfloor n/2 \rfloor$ 次

行3--4 求 max 和 min 比较:

至多 $2\lceil n/2 \rceil - 2$ 次

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

$$= n + \lceil n/2 \rceil - 2$$

$$= \lceil 3n/2 \rceil - 2$$

分治算法

1. 将数组 L 从中间划分为两个子数组 L_1 和 L_2
2. 递归地在 L_1 中求最大 max_1 和 min_1
3. 递归地在 L_2 中求最大 max_2 和 min_2
4. $max \leftarrow \max\{ max_1, max_2 \}$
5. $min \leftarrow \min\{ min_1, min_2 \}$

最坏情况时间复杂度

假设 $n = 2^k$,

$$W(n) = 2W(n/2) + 2$$

$$W(2) = 1$$

解
$$\begin{aligned} W(2^k) &= 2W(2^{k-1}) + 2 \\ &= 2[2W(2^{k-2}) + 2] + 2 \\ &= 2^2W(2^{k-2}) + 2^2 + 2 = \dots \\ &= 2^{k-1} + 2^{k-1} + \dots + 2^2 + 2 \\ &= 3 \cdot 2^{k-1} - 2 = 3n/2 - 2 \end{aligned}$$

选择算法小结

选最大：顺序比较, 比较次数 $n-1$

选最大最小

- 选最大+ 选最小, 比较次数 $2n-3$
- 分组： 比较次数 $\lceil 3n/2 \rceil - 2$
- 分治： $n=2^k$, 比较次数 $3n/2-2$