

Median and Order Statistics

Order Statistics 問題敘述

- 在 n 個元素中，找出其中第 i 小的元素。
 - $i = 1$ ，即為找最小值。
 - $i = n$ ，即為找最大值。
 - $i = \lfloor n+1/2 \rfloor$ 或 $\lceil n+1/2 \rceil$ ，即為找中位數。
- 直覺解法：排序，然後輸出第 i 個元素。
需要 $O(n \log n)$ 的時間。

9.1 尋找最小值 (最大值)

Minimum(A)

$\text{min} = A[1]$

for $i = 2$ **to** n **do**

if $\text{min} > A[i]$ **then** $\text{min} = A[i]$

return min

- 所花的時間 $T(n)=O(n)$ (恰使用 $n-1$ 比較)
- $n-1$ 次比較是最佳解 (一次比較只能確定一個元素不是最小值)
- 找最大值可以使用類似的方法。

同時找尋最大數最小數

- 比較 $2n - 2$ 次。
- 是否可以使用更少次的比較找到？
- $3\lfloor \frac{n}{2} \rfloor$ 次比較即可，Why?

找尋中位數

- 如反覆套用尋找最小值的演算法，找出第 i 小的元素將花 $O(in)$ 的時間。
- 故套用到找中位數的時候，需要花 $O(n^2)$ 的時間。比排序花的還要多。
- 是否能找到一個演算法能在 $O(n)$ 的時間內找到中位數呢？

9.2 隨機演算法

```
RANDOMIZED-SELECT( $A, p, r, i$ )
1  if  $p == r$ 
2      return  $A[p]$           //  $1 \leq i \leq r - p + 1$  when  $p == r$  means that  $i = 1$ 
3   $q = \text{RANDOMIZED-PARTITION}(A, p, r)$ 
4   $k = q - p + 1$ 
5  if  $i == k$ 
6      return  $A[q]$           // the pivot value is the answer
7  elseif  $i < k$ 
8      return  $\text{RANDOMIZED-SELECT}(A, p, q - 1, i)$ 
9  else return  $\text{RANDOMIZED-SELECT}(A, q + 1, r, i - k)$ 
```

pivot

$$\text{helpful: } |A^{(j)}| \leq 3/4 |A^{(j-1)}|$$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(0)}$	6	19	4	12	14	9	15	7	8	11	3	13	2	5	10

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(1)}$	6	4	12	10	9	7	8	11	3	13	2	5	14	19	15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(2)}$	3	2	4	10	9	7	8	11	6	13	5	12	14	19	15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(3)}$	3	2	4	10	9	7	8	11	6	12	5	13	14	19	15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(4)}$	3	2	4	5	6	7	8	11	9	12	10	13	14	19	15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A^{(5)}$	3	2	4	5	6	7	8	11	9	12	10	13	14	19	15

p	r	i	partitioning	helpful?
-----	-----	-----	--------------	----------

1	15	5		
---	----	---	--	--

			1	no
--	--	--	---	----

1	12	5		
---	----	---	--	--

			2	yes
--	--	--	---	-----

4	12	2		
---	----	---	--	--

			3	no
--	--	--	---	----

4	11	2		
---	----	---	--	--

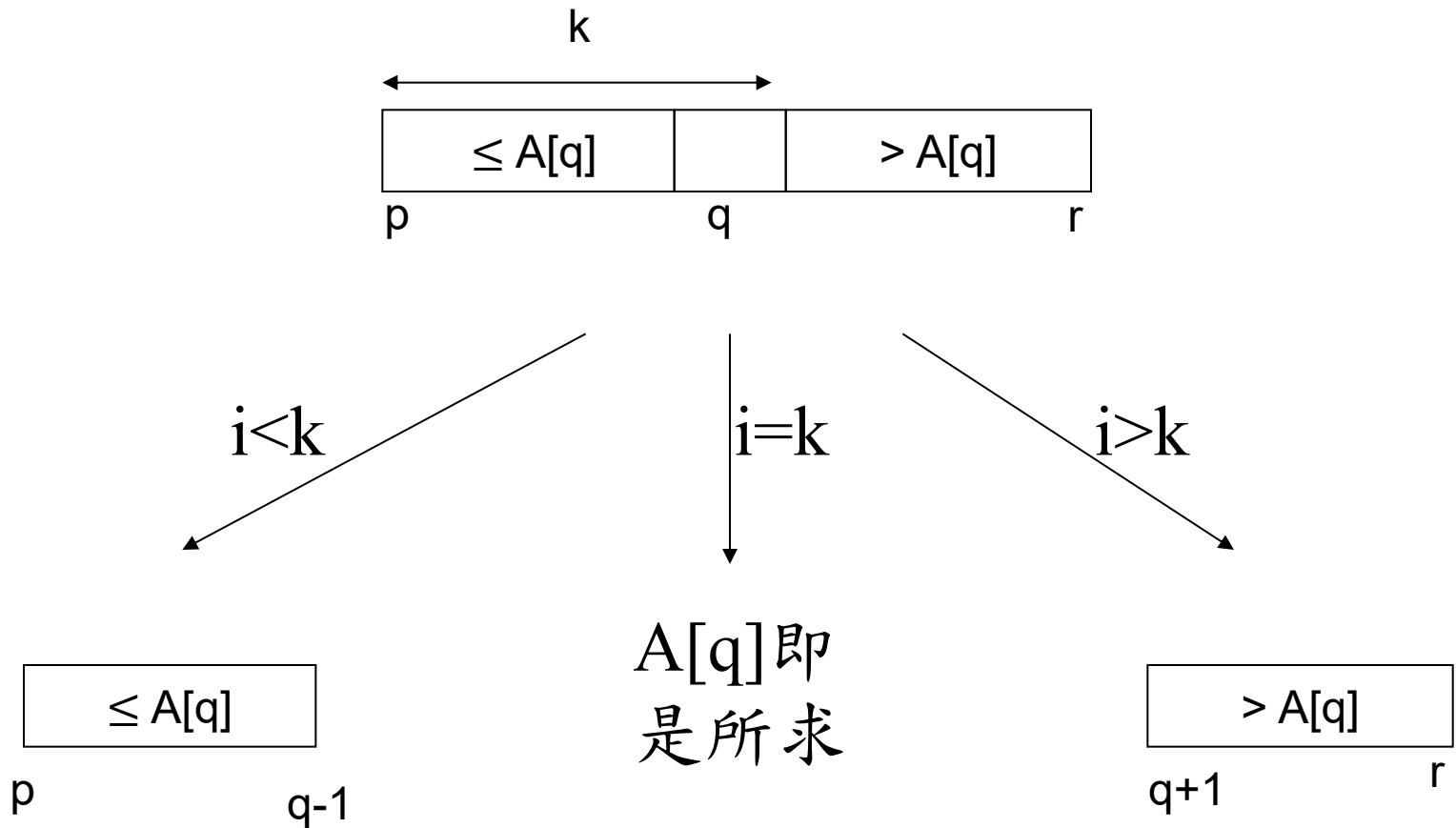
			4	yes
--	--	--	---	-----

4	5	2		
---	---	---	--	--

			5	yes
--	--	--	---	-----

5	5	1		
---	---	---	--	--

Median and Order Statistics



在此找第 i 大的元素

在此找第 $i-k$ 大的元素

分析

- 幸運的例子：每次都能去除十分之一以上。

$$T(n) = T(\frac{9n}{10}) + \Theta(n) = O(n)$$

可利用第四章的Master法計算出。

$$n^{\log_{10} 1} = n^0 = 1.$$

- 運氣不好的例子：每次都只能去除一個元素。

$$T(n) = T(n-1) + \Theta(n) = O(n^2)$$

- 平均計算：
為了求上限方便起見，假定第*i*小的元素總是掉在較大的Partition中。
- 對任一 $k=1..n$ ， $A[p..q]$ 恰有 k 個元素的機率為 $1/n$ 。
- 令 $X_k = I\{A[p..q] \text{ 恰有 } k \text{ 個元素}\}$ ，---Indicator random variable。
- $E[X_k] = 1/n$ 。

$$\begin{aligned}
 T(n) &\leq \sum_{k=1}^n X_k \cdot (T(\max(k-1, n-k)) + O(n)) \\
 &= \left[\sum_{k=1}^n X_k \cdot T(\max(k-1, n-k)) \right] + O(n)
 \end{aligned}$$

$$\begin{aligned}
E[T(n)] &\leq E\left[\sum_{k=1}^n X_k \cdot T(\max(k-1, n-k))\right] + O(n) \\
&= \sum_{k=1}^n E[X_k \cdot T(\max(k-1, n-k))] + O(n) \\
&= \sum_{k=1}^n E[X_k] \cdot E[T(\max(k-1, n-k))] + O(n) \\
&= \sum_{k=1}^n \frac{1}{n} \cdot E[T(\max(k-1, n-k))] + O(n)
\end{aligned}$$

$$\because \max(k-1, n-k) = \begin{cases} k-1 & \text{if } k > \lceil n/2 \rceil \\ n-k & \text{if } k \leq \lceil n/2 \rceil \end{cases}$$

$$E[T(n)] \leq \frac{2}{n} \sum_{k=\lceil n/2 \rceil}^{n-1} E[T(k)] + O(n).$$

解 $E[T(n)] \leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} E[T(k)] + O(n)$

利用代換法：假定 $E[T(n)] \leq cn$ 。

$$\begin{aligned}
 E[T(n)] &\leq \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} ck + an \leq \frac{2c}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} k + an \\
 &= \frac{2c}{n} \left(\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2 \rfloor - 1} k \right) + an \\
 &= \frac{2c}{n} \left(\frac{n(n-1)}{2} - \frac{1}{2} (\lfloor n/2 \rfloor - 1) \lfloor n/2 \rfloor \right) + an \\
 &\leq \frac{2c}{n} \left(\frac{n(n-1)}{2} - \frac{1}{2} (n/2 - 2)(n/2 - 1) \right) + an \\
 &= c \left(\frac{3n}{4} + \frac{1}{2} - \frac{2}{n} \right) + an \leq c \left(\frac{3n}{4} + \frac{1}{2} \right) + an = cn - \left(\frac{cn}{4} - \frac{c}{2} - an \right)
 \end{aligned}$$

可以取足夠大的 c 使得 $c(n/4 - 1/2)$ 大於 an 使得最後一個不等式成立。

9.3 Worst case linear-time order statistics

- 理論研究上的興趣。
- 關鍵的想法：找到一個可以產生良好分割效果的元素 x 。即大於 x 及小於 x 的元素個數不至於太少。

Select(i)

{

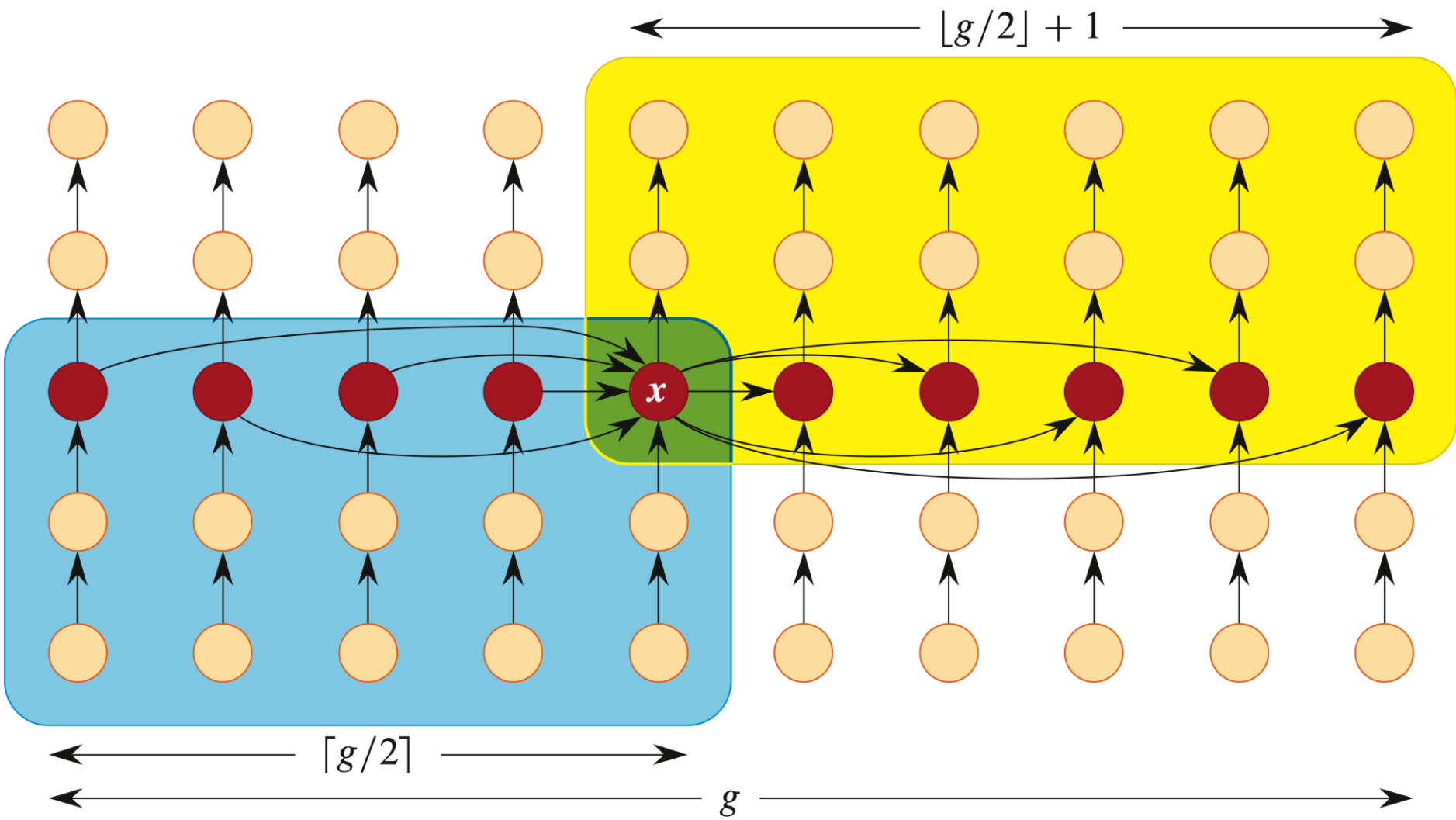
1. 將 n 個元素分成 $\lfloor n/5 \rfloor$ 個由5個元素構成的小群
2. 找出每群的中位數
3. 利用Select函數找出這 $\lfloor n/5 \rfloor$ 個中位數的中位數 x
4. 利用 x 作為Partition使用的Pivot，Partition後 $x=A[k]$
5. 如 $i=k$ 則return x
6. 如 $i < k$ 則對小於 x 的那些元素做Select(i)
7. 如 $i > k$ 則對大於 x 的那些元素做Select(i-k)

}

```

SELECT(A, p, r, i)
1  while (r − p + 1) mod 5 ≠ 0
2      for j = p + 1 to r                                // put the minimum into A[p]
3          if A[p] > A[j]
4              exchange A[p] with A[j]
5          // If we want the minimum of A[p : r], we're done.
6          if i == 1
7              return A[p]
8          // Otherwise, we want the (i − 1)st element of A[p + 1 : r].
9          p = p + 1
10         i = i − 1
11     g = (r − p + 1)/5                                    // number of 5-element groups
12     for j = p to p + g − 1                            // sort each group
13         sort {A[j], A[j + g], A[j + 2g], A[j + 3g], A[j + 4g]} in place
14     // All group medians now lie in the middle fifth of A[p : r].
15     // Find the pivot x recursively as the median of the group medians.
16     x = SELECT(A, p + 2g, p + 3g − 1, ⌈g/2⌉)
17     q = PARTITION-AROUND(A, p, r, x) // partition around the pivot
18     // The rest is just like lines 3–9 of RANDOMIZED-SELECT.
19     k = q − p + 1
20     if i == k
21         return A[q]                                     // the pivot value is the answer
22     elseif i < k
23         return SELECT(A, p, q − 1, i)
24     else return SELECT(A, q + 1, r, i − k)

```



Median and Order Statistics

分析

- 由上頁圖示，可知至少有 $3\left(\left\lceil\frac{1}{2}\left\lceil\frac{n}{5}\right\rceil\right\rceil-2\right) \geq \frac{3n}{10}-6$ 的元素較 x 來的大。
- 同理，至少有 $3n/10 - 6$ 的元素較 x 來的小。
- 如果Partition過， $i \neq k$ ，則至多只要在 $7n/10+6$ 個元素的情況下遞迴執行Select。
- 而先前找出 $\lceil n/5 \rceil$ 小群中位數的中位數時，只在 $n/5$ 個元素的情況下遞迴執行 Select。
- 故 $T(n) \leq T(\lceil n/5 \rceil) + T(7n/10+6) + \Theta(n)$, for $n \geq 140$.

分析

- 利用替换法，令 $T(n) \leq cn$

$$\begin{aligned} T(n) &\leq c \lceil n/5 \rceil + c(7n/10 + 6) + an \\ &\leq cn/5 + c + c7n/10 + 6c + an \\ &= 9cn/10 + 7c + an \\ &= cn + (-cn/10 + 7c + an) \\ &\leq cn, \quad \text{if } -cn/10 + 7c + an \leq 0! \end{aligned}$$

Worst case linear-time order statistics之應用

- 可用於實作時間複雜度為 $\Theta(n \log n)$ 的Quicksort。
- Modified-Quicksort
 - {
 - 利用Select找出中位數 x
 - 使用 x 作為Pivot進行Partition
 - 將大於 x 以及小於 x 的兩部分遞迴執行排序}
- 由於使用中位數進行Partition，所以大於 x 及小於 x 兩部份均不大於 $n/2$ ，故 $T(n)=\Theta(n \log n)$ 。