Order Statistics

Find i^{th} minimum element in an array of n- elements (distinct)

- Finding Minimum
- Finding Maximum
- Finding Median
- General i (Find i^{th} minimum)

Finding Minimum/Maximum

- Comparison Sort ($\Omega(n \log n)$)

Can you do better?

10	35	20	5	4	60	68	23

$O(n)$

$\boxed{\cancel{10}\ \cancel{5}\ 4}$

min

$\hookrightarrow$ min

- [ ] Min # of Comparisons you need? (min) $(n-1)$ $\Longrightarrow$ Tournament method (Each comparison $\rightarrow$ one element discarded from Min)
- [ ] Min # - - - - - max? $(n-1)$

Finding Simultaneous Minimum & Maximum

Given an array A of n distinct integers, report the minimum & maximum in "least no. of comparisons".

10	35	20	5	4	29	35	26

	Min	Max
	10	10
35/Max	10	35
20/Min, 20/Max	10	35
5/Min,	5	

element < min
update(min)
no more comparison

O/W
if element > max
update(max)

① **Naive**:

- Min (n−1)
  Max (n−1)

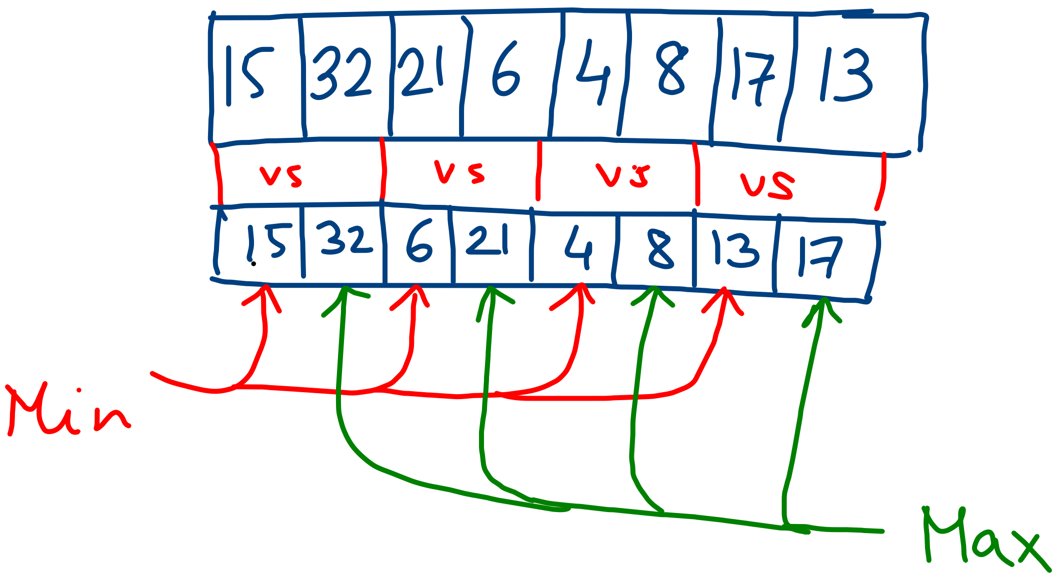
$$2n-2$$

② **Worst-Case**:

First element min

Total Comparisons = (2n−2)

(3)



For each element at most 1 type-I comparison

$\hookrightarrow \frac{n}{2}$ (type I comp)

How can you show this is the minimum no. of comparisons required?

- Comparison type I: Discard min & max set by 1 $(n/2)$
- Comparison type II: Discard min set by 1
- – – – III: – – –

$(n-2)$ type II + III

\# Comparisons

$$= \frac{n}{2} + \left(\frac{n}{2} - 1\right) + \left(\frac{n}{2} - 1\right)$$

$$\approx \left(\frac{3n}{2} - 2\right)$$

Finding 2^{nd} Minimum

30	15	6	17	5	4	23	17

15 30 6 17 4 5 17 23

15 6 4 17

$$\left(\frac{3n}{2} - 2\right)$$

$n/2$

Can you reduce it?

$$Min^m(15, 6, 4, 17) \to 4 \Rightarrow n/2 - 1$$

$$Min^m(15, 6, 17, 5) \Rightarrow n/2 - 1$$

$(\checkmark)$

Given an array of size n, find an element which is not (i) maximum / (ii) 3rd maximum

- One Comparison ($A[1]$ vs $A[2]$) } Report the minimum

- $A[1]$ vs $A[2]$ → Take first 4 elements
  $A[2]$ vs $A[3]$ → Report min of them
  (4) (?) $A[1]$ vs $A[3]$
  $A[1]$ vs $A[4]$

$A[1]$ min^m

$A[3]$ min^m $A[1] < A[3] < A[2]$

$A[1]$ min^m

Comp = 4 − 1 = 3

$A[4]$ min^m ⇒ Report $A[4]$

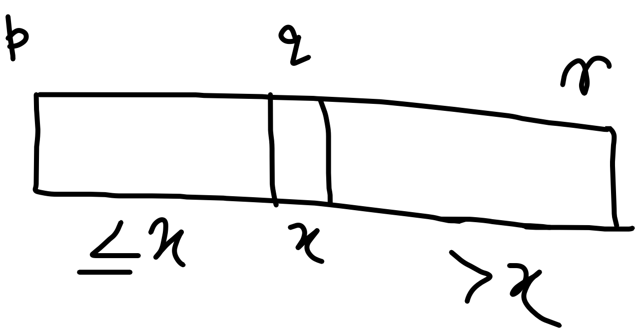
$A[1]$ — ⇒ - —

Finding i^{th} Minimum

- Sort the Array
- Report the i^{th} element

$\Theta(n \log n)$

Can you do better?



Worst Case

$O(n \log n)$

Select(A, p, r, i)

$A[p..r]$ find i^{th} minimum

q = Randomized_Partition(A, p, r) // $\Theta(n)$

if ($i == q - p + 1$)
    return $A[q]$
else if ($i < q - p + 1$)
    Select(A, p, q-1, i)
else
    Select(A, q+1, r, i - q + p - 1)

Avg Case

$X_k \to \{ I_{A[k..n] \text{ returns } k} \}$

$E[X_k] = 1 \cdot \frac{1}{n} + 0 \cdot \left(1 - \frac{1}{n}\right) = \frac{1}{n}$

$$E(T(n)) \leq E\left(\sum_{k=1}^{n} X_k \cdot \max\{T(k-1), T(n-k)\} + O(n)\right)$$

$$\leq E\left(2\sum_{k=\frac{n}{2}}^{n} X_k \cdot T(k)\right) + O(n)$$

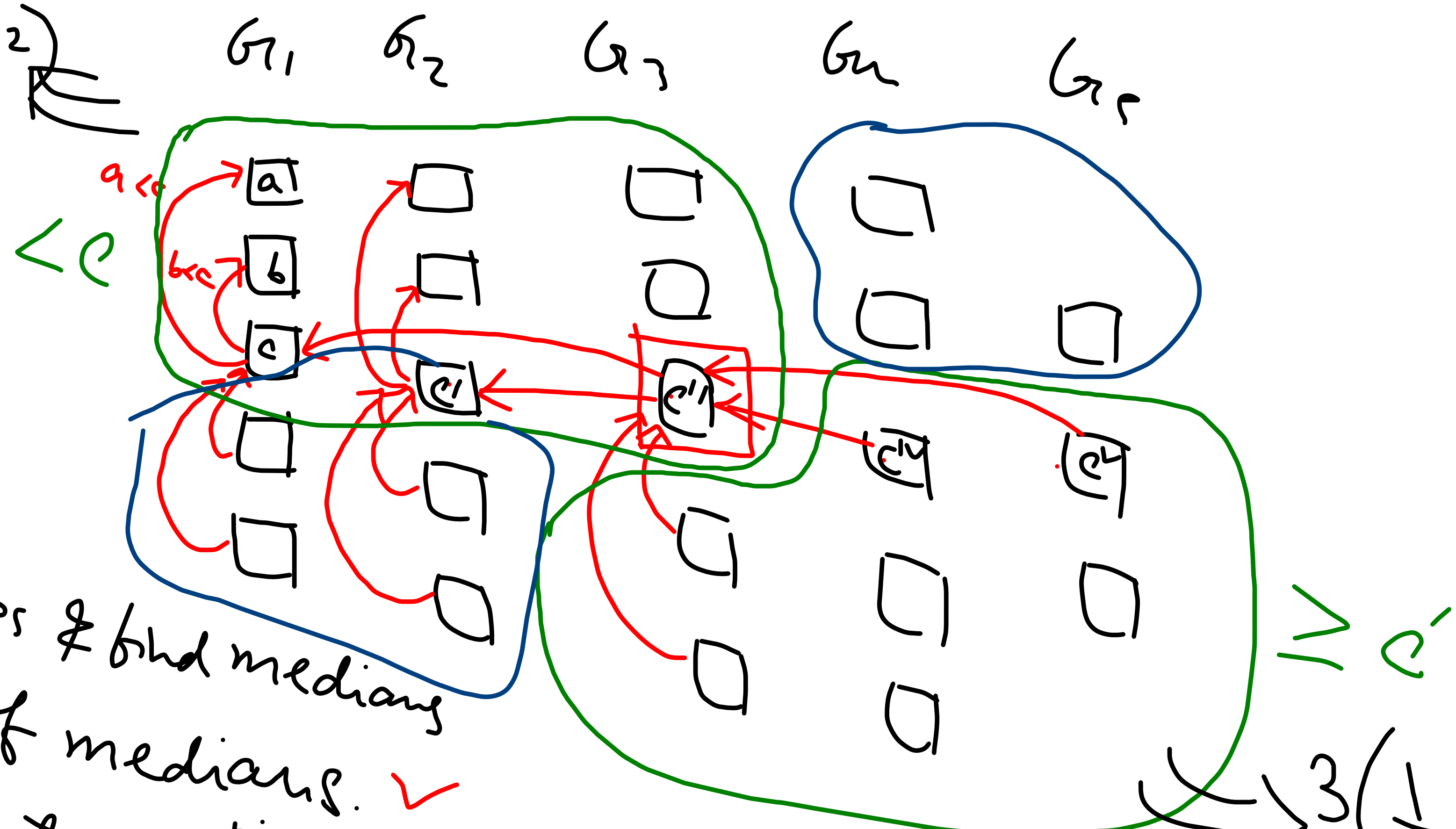
Recurrence by Substitution method

$$= \frac{1}{n}\sum_{k=n/2}^{n} E(T(k)) + O(n) \approx O(n)$$

Worst Case $O(n)$ Selection

Partition carefully so that it is not a skewed one

$\geq 3\left(\frac{1}{2}\cdot\frac{n}{5}-2\right)$



$\geq 3\left(\frac{1}{2}\cdot\frac{n}{5}-2\right)$

Idea:

- Make $\left\lceil \frac{n}{5} \right\rceil$ groups of 5 elements
- Sort each Groups & find medians
- Find median of medians. ✓
- Use this element as pivot in the partition.

$$T(n) = \begin{cases} T\left(n - \frac{3n}{10}\right) + O(n) + T\left(\left\lceil \frac{n}{5} \right\rceil\right) & \text{for } n \geq 140 \\ = O(1), \text{ for } n < 140 \end{cases}$$

$n < 140 \rightarrow$ Use any simple sorting.

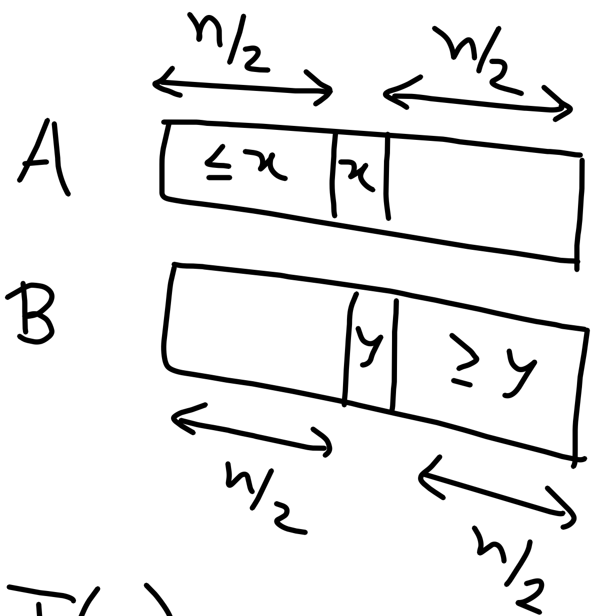
$O(1)$

$T(n) = O(n)$

Try to group with
(i) 3 elements,
(ii) 7 elements, and
Check whether it works

Ex

1. Find median of 2 sorted Arrays in $O(\log n)$ time.



Median$(A[1..n], B[1..n])$

---

if $(x \le y)$

Median$(A[n/2..n], B[1..n/2])$

else

Median$(B[n/2..n], A[1..n/2])$

$$T(n) = T(n/2) + 1$$

$$T(n) = O(\log n)$$