Medians and Order Statistics

Given a set of n elements, the i^{th} order statistic is the i^{th} smallest element in the set.

minimum – first order statistic $(i=1)$
maximum – n^{th} order statistic $(i=n)$

Median

- n is odd

the median is the i^{th} order statistic for $i = \frac{n+1}{2}$

example

1	2	3	4	5
9	2	14	6	1

A

$n=5$

$i = \frac{5+1}{2} = 3$

the median $= 3^{rd}$ order statistic $= 6$

- n is even

the lower median is the i^{th} order statistic for $i = \frac{n}{2}$

the upper median is the i^{th} order statistic for $i = \frac{n}{2} + 1$

example

1	2	3	4	5	6
9	2	14	6	1	20

A

$n=6$

the lower median $= 3^{rd}$ order statistic $= 6$

the upper median $= 4^{th}$ order statistic $= 9$

The selection problem

Input: A set A of n (distinct) numbers and an integer i with $1 \le i \le n$

Output: The i^{th} order statistic element of A.

Minimum and maximum

- we can compute minimum (or maximum) using $n-1$ comparisons

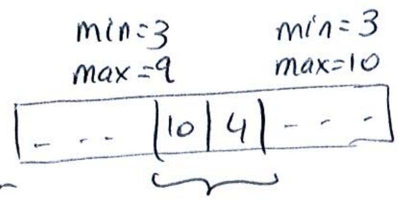
MINIMUM(A)

```
min = A[1]
for i = 2 to A.length
    if min > A[i]
        min = A[i]
return min
```

- used $n-1$ comparisons, where $n = A.length$

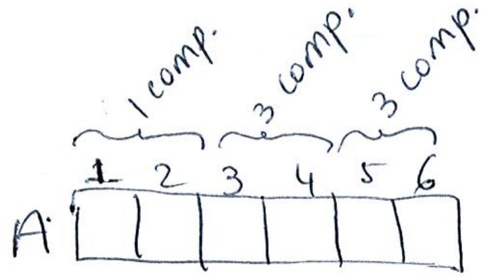
Simultaneous minimum and maximum

- using the previous method requires $2n-2$ comparisons
- solution that requires at most $3\lfloor \frac{n}{2} \rfloor$ comparisons
  - maintain the minimum and maximum elements seen thus far
  - process the elements in pairs:
    - compare the two elements with each other
    - compare the smaller with the min so far
    - compare the larger with the max so far



3 comparisons

- initialization
  - n is even

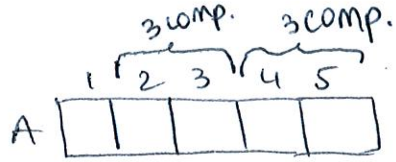


1 comparison
- compare $A[1]$ and $A[2]$
- assign
  - min = the smaller value
  - max = the larger value
- then process elements in pairs

- n is odd

  $min = max = A[1]$

  then process elements in pairs



- number of comparisons

  - n is even $\quad 1 + \left(\frac{n-2}{2}\right) \cdot 3 = 1 + 3\frac{n}{2} - 3 = 3\frac{n}{2} - 2 < 3 \cdot \lfloor \frac{n}{2} \rfloor$

  - n is odd $\quad \left(\frac{n-1}{2}\right) \cdot 3 = 3 \lfloor \frac{n}{2} \rfloor$

In both cases the total number of comparisons is at most $3\lfloor \frac{n}{2} \rfloor$.

The selection problem

The selection problem seeks to find the i^{th} order statistic in a set A of n distinct numbers, where $1 \le i \le n$.

- Simple solution with $RT = O(n \cdot \lg n)$
  - sort the numbers using heapsort or merge sort in $O(n \lg n)$
  - return $A[i]$

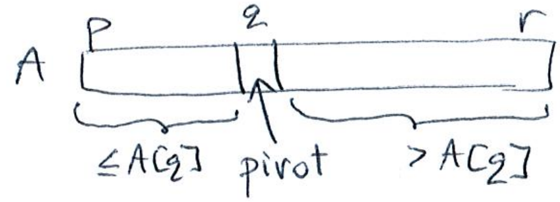
- Solution in expected linear time
  - the algorithm is similar to quicksort, but it recurses on only one side of the partition
  - uses divide-and-conquer

general problem: find the i^{th} order statistic in $A[p..r]$

Divide

call RANDOMIZED-PARTITION to partition $A[p..r]$ into two subarrays $A[p..q-1]$ and $A[q+1..r]$ such that

- elements in $A[p..q-1]$ are $\le A[q]$
- elements in $A[q+1..r]$ are $> A[q]$



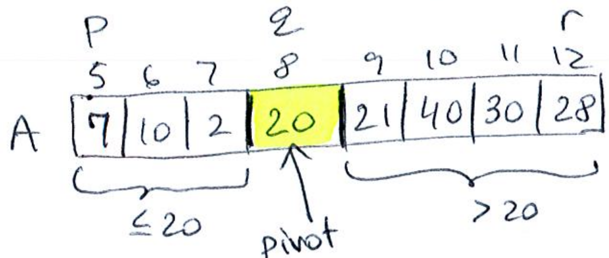
Conquer

$k = q - p + 1$

$A[q]$ is the k^{th} smallest element

$\Rightarrow A[q]$ is the k^{th} order statistic

- if $i = k$, then return $A[q]$
- if $i < k$, then find the i^{th} order statistic in $A[p..q-1]$
- if $i > k$, then find the $(i-k)^{th}$ order statistic in $A[q+1..r]$



$k = 8 - 5 + 1 = 4$

$A[q]$ is the 4^{th} order statistic

Combine

- nothing to be done

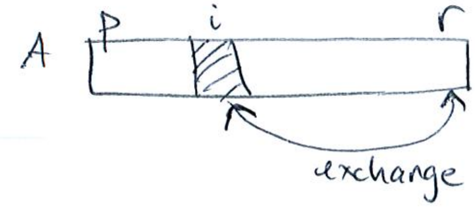
```
RANDOMIZED-SELECT(A, p, r, i)
if p == r
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q - p + 1
if i == k
    return A[q]
elseif i < k
    return RANDOMIZED-SELECT(A, p, q-1, i)
else // i > k
    return RANDOMIZED-SELECT(A, q+1, r, i-k)
```

- initial call: RANDOMIZED-SELECT(A, 1, A.length, i)

<u>RANDOMIZED-PARTITION (A, p, r)</u>

i = RANDOM (p, r)
exchange A[r] with A[i]
$\Theta(n)$ return PARTITION (A, p, r)

RT of RANDOMIZED-PARTITION is $\Theta(n)$



<u>RT analysis of randomized-select</u>

- <u>Worst-case</u>

- the two subproblems are completely unbalanced:
  - one subproblem has $(n-1)$ elements
  - one subproblem has 0 elements

and the algorithm recurses on the larger subproblem

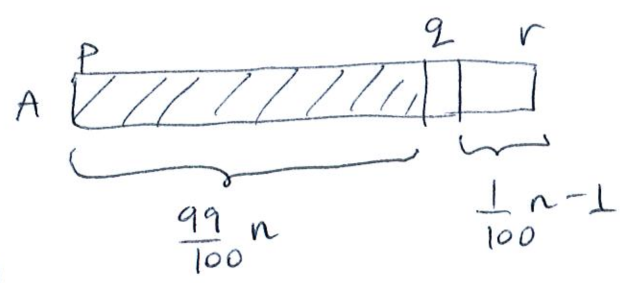
$$T(n) = T(n-1) + \Theta(n)$$

$\uparrow$ RT for RANDOMIZED-PARTITION

$T(n) = T(n-1) + cn$ (solved previously)

$$\boxed{T(n) = \Theta(n^2)}$$

- <u>Best-case</u>



$$T(n) = T\left(\frac{99}{100}n\right) + \Theta(n)$$

$\uparrow$ RT for RANDOMIZED-PARTITION

$$T(n) = T\left(\frac{99}{100}n\right) + cn$$

$\boxed{T(n) = \Theta(n)}$ (case 3 of the Master Theorem)