ORDER STATISTICS - MEDIAN FINDING, RANK SELECTION

Given $n$ unsorted elements, find the $k$-th smallest.

We will assume distinct elements.

$\Rightarrow$ easy $O(n)$ if $k=O(1)$ or $n=O(1)$.

(median is hardest)

Algorithm by: Blum, Floyd, Pratt, Rivest, Tarjan

1973
Let this run in time $T(n)$

$\text{Select}(r, 1...n)$  // find $r^{th}$ smallest # within array[1...n]

1) Form $\frac{n}{5}$ groups of 5 elements  // the last group can have $\leq$ 5
2) Find median in each group  // brute force.
3) Recursively find $x =$ median-of-medians
4) Compare all elements to $x$  $\rightarrow$ compute $\text{rank}[x] = p$
5) if $\text{rank}[x] = p = r$ , done, Else use $x$ as pivot to partition input
   (set up binary search)
6) if $p > r$  // $\text{rank}[x] > r$, so search lower
   Select($r, 1...p-1$)
   else // $p < r$, so search higher
   Select($r-p, p+1...n$)
1) Form \( \frac{n}{5} \) groups of 5 elements \( \Theta(n) \) ... in fact, no work

Don't worry about extras could add three elements = \( \infty \)

OR

remove MAX & MAX-1 \( \Theta(n) \)
1) Form \( \frac{n}{5} \) groups of 5 elements \( \Theta(n) \)
2) Find median in each group

\[ \frac{n}{5} \cdot \Theta(1) = \Theta(n) \]
1) Form $\frac{n}{5}$ groups of 5 elements $\Theta(n)$

2) Find median in each group (and re-organize) $\frac{n}{5} \cdot \Theta(1) = \Theta(n)$
1) Form $\frac{n}{5}$ groups of 5 elements $\Theta(n)$
2) Find median in each group (and re-organize) $\frac{n}{5} \cdot \Theta(1) = \Theta(n)$
3) Recursively find $X = \text{median-of-medians}$

Time $\rightarrow T(\frac{n}{5})$
1) Form \( \frac{n}{5} \) groups of \( 5 \) elements \( \Theta(n) \)

2) Find median in each group (and re-organize) \( \frac{n}{5} \cdot \Theta(1) = \Theta(n) \)

3) Recursively find \( x = \text{median-of-medians} \) (and re-organize) \( T\left( \frac{n}{5} \right) + \Theta(n) \)
1) Form $\frac{n}{5}$ groups of 5 elements $\Theta(n)$

2) Find median in each group (and re-organize) $\frac{n}{5} \cdot \Theta(1) = \Theta(n)$

3) Recursively find $X = \text{median-of-medians}$ (and re-organize) $T\left(\frac{n}{5}\right) + \Theta(n)$

Re-organizing is not part of the algorithm. It's part of the proof. (although we could afford it)

That's the algorithm. Now to find $T(n)$
Let \( x \rightarrow y \) mean \( x > y \).
\[ \#\text{"big"} = \#\text{"small"} \geq 3 \cdot \frac{n/5}{2} \leq n \]

- Columns containing big elements
- Big items per column
\[ \# \text{big} \geq 3 \cdot \frac{n/3}{\lfloor 2 \rfloor} \geq 3 \cdot \frac{n}{10} \]

\[ \frac{n}{\lfloor s \rfloor} \rightarrow \frac{n}{s} \text{ if we ignore incomplete column} \]

\[ \frac{n/2}{L^2} \rightarrow \frac{n/2}{2} \text{ if } n: \text{even} \]

\( \Theta(n) \text{ work takes care of this} \)
$\#^{\text{big}} = \#^{\text{small}} \geq 3 \cdot \frac{n/5}{\sqrt{2}} \geq 3 \cdot \frac{n}{10}$

$\geq \frac{1}{4} n$ For $n \geq 50$

If $x$ is not at the target rank/index, and we need to search lower (i.e., $\text{rank}(x) > \text{target}$), then recurse on all elements except "big" [symmetrically, if searching for $\text{target} > \text{rank}(x)$, recurse on all except "small"]

```
| r | x | . . | . . . . . . . . |
```

recurse ← [ignore]
If \( \times \) is not at the target rank/index, and we need to search lower (i.e., \( \text{rank}(x) > \text{target} \)), then recurse on all elements except "big".

Symmetrically, if searching for \( \text{target} < \text{rank}(x) \), recurse on all except "small".

\[
T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{3n}{4}\right) + \Theta(n) \quad \text{steps 1 & 2: find medians of 5 & partition}
\]

\[\text{find } x \quad \text{recurse if } \text{rank}(x) \neq \text{target}\]

\[\text{for } n > 50 \text{ and } \#	ext{"big"} = \#	ext{"small"} \geq 3 \cdot \frac{n^{7/8}}{\sqrt{2}} \geq 3 \cdot \frac{n}{10} \geq \frac{1}{4} n\]

For \( n \geq 50 \)
\[ T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{3n}{4}\right) + \Theta(n) \]

Claim \( T(n) \leq c \cdot n \)

\[ \leq c \cdot \frac{n}{5} + c \cdot \frac{3n}{4} + dn \]

\[ = \frac{19}{20} cn + dn = cn - \left(\frac{1}{20} cn - dn\right) \leq cn \text{ if } c > 20d \]

QED
collect medians-of-5
solve new problem
(return median)
will rank somewhere in middle 50% of original list
\( \langle T(\frac{3n}{4}) \rangle \)

Find \( \text{rank}(x) \)

If \( x \neq \text{target} \), recurse on \( \langle \frac{3n}{4} \rangle \) of list
What were they thinking? (my guess)

- Goal: $\Theta(n)$  
  $[\Omega(n)$ lower bound; $O(n\log n)$ is trivial]

- Exploit $\sim$ geometric series:  
  $T(n) = T\left(\frac{n}{b}\right) + O(n)$

  or $T(n) = T(xn) + T(yn) + O(n)$

  ... where $x+y < 1$

$\Rightarrow$ spend $T(xn) + O(n)$ time

  to make sure that only $yn$ candidates remain
Why groups of 5?

In class I showed that groups of 3 no longer give $O(n)$ and mentioned that groups must have constant size to keep $O(n)$. After that, it's an optimization to get a better leading constant.