ORDER STATISTICS - MEDIAN FINDING, RANK SELECTION

Given \( n \) unsorted elements, find the \( k \)-th smallest.

We will assume distinct elements.

\( \Leftrightarrow \) 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 $T(n)$ we are looking for the element w/ rank $r$

$\text{Select}(r, 1...n)$ // 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 $x = \text{median-of-meds}$
4) Compare all elements to $x$ $\rightarrow$ compute rank[$x$] = $p$
5) if rank[$x$] = $p = r$, done, Else use $x$ as pivot to partition input (set up binary search)
6) if $p > r$ // rank[$x$] $> r$, so search lower
   \hspace{1cm} \text{Select}(r, 1...p-1)$

   else // $p < r$, so search higher
   \hspace{1cm} \text{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 $x = \text{median-of-median}$

```
1) 2
```

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) \( T(\frac{n}{5}) + \Theta(n) \)
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) $T(\frac{n}{5}) + \Theta(n)$

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 $X > Y$
\[
\# \text{"big"} = \# \text{"small"} \geq 3 \cdot \left\lfloor \frac{n/5}{2} \right\rfloor
\]

columns containing big elements

big items per column
\[ \# \text{"big"} = \# \text{"small"} \geq 3 \cdot \frac{\frac{n/3}{2}}{10} \geq 3 \cdot \frac{n}{10} \]

\[ \frac{n}{L^5} \rightarrow \frac{n}{5} \text{ if we ignore incomplete column} \]

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

\[ \Theta(n) \text{ work} \]

\[ \text{takes care of this} \]
\[ \#\text{big} = \#\text{small} \geq 3 \cdot \frac{\lceil \frac{n}{5} \rceil}{2} \geq 3 \cdot \frac{n}{10} \geq \frac{1}{4}n \text{ for } n \geq 50 \]

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 \( \text{big} \). [Symmetrically, if searching for \( \text{target} > \text{rank}(x) \), recurse on all except \( \text{small} \)].
"big" = "small" ≥ 3 \cdot \frac{n/5}{2} ≥ 3 \cdot \frac{n}{10} \geq \frac{1}{4}n \quad \text{For } n \geq 50

if \( \times \) is not at the target rank/index, and we need to search lower (i.e., rank(x) > target), then recurse on all elements except "big"

[ symmetrically, if searching for target > rank(x), recurse on all except "small"

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

find x

recurse if rank(x) ≠ target

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

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
$T\left(\frac{n}{5}\right)$

solve new problem
(return median)
will rank somewhere in middle 50% of original list
\[ T \left( \frac{3n}{4} \right) \]

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

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

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

- Exploit \~ geometric series: 
  \[T(n) = T(\frac{n}{b}) + O(n)\]
  or \[T(n) = T(xn) + T(yn) + O(n)\]
  ... where \(x+y < 1\)

\[\Rightarrow\text{ spend } T(xn) + O(n) \text{ time}\]

to make sure that only \(yn\) candidates remain