Median:
- Given an unsorted list \( A = [a_1, \ldots, a_n] \) of \( n \) numbers, find the median of \( A \).
  (for concreteness, say the median is the \( \frac{n-1}{2} \)-th smallest)

Useful to solve the following more general problem:
- Given unsorted \( A \) & integer \( k \) where \( 1 \leq k \leq n 
  find the \( k \)-th smallest of \( A \).

Easy alg.: Sort \( A \) & output \( k \)-th element of sorted array.
\( \Rightarrow \mathcal{O}(n \log n) \) time.

This lecture: \( \mathcal{O}(n) \) time algorithm due to
[Blum, Floyd, Pratt, Rivest, & Tarjan '73]

D&C approach reminiscent of Quicksort:
Quicksort:  - Choose pivot \( p \)
  - Recursively sort \( A < p \) & \( A > p \).
Since searching instead of sorting we only need to consider 1 subproblem.

**QuickSelect** *(A,k)*:

1. Choose a pivot *p* (How? This is the challenging step)
2. Partition A into *A*_\(<p*, *A*_\=p*, & *A*_\>p*.
3. If \(|A\<p*| < k \leq |A\<p*| + |A\=p*| \) then
   
   return **QuickSelect** *(A\<p*, k)*

   If \(|A\<p*| < k \leq |A\<p*| + |A\=p*| \) then

   return *(p)*

   If \(|A\<p*| + |A\=p*| < k \) then

   return **QuickSelect** *(A\>p*, k-|A\<p*|-|A\=p*|)*

---

**Example:** \(A = [5,2,20,17,11,13,8,9,11]\)

Say *p* = 11, \(A\<p* = [5,2,8,9]\), \(A\=p* = [11,11]\), \(A\>p* = [20,17,13]\)

if \(k \leq 4\) then we know it's \(k^{th}\) smallest in \(A\<p*\)
if \(k = 5\) or \(6\), then we know it's here
if \(k > 6\), then we want \((k-6)^{th}\) smallest in
Want $O(n)$ running time so aim for a recurrence such as:

$$T(n) = T\left(\frac{n}{2}\right) + O(n)$$

But note

$$T(n) = T(0.99n) + O(n)$$

also solves to

$$T(n) = O(n)$$

For any constant $a < 1, T(n) = T(an) + O(n) = O(n)$.

We'll create 2 subproblems of sizes $an$ & $bn$ where $a+b < 1$ and use that

$$T(n) = T(an) + T(bn) + O(n) = O(n)$$

when $a+b < 1$.

Our goal is to find a good pivot meaning:

P is a good pivot if $|A_p| \leq \frac{3}{4} n$ & $|A_{\bar{p}}| \leq \frac{3}{4} n$.

If we can find a good pivot in $O(n)$ time, then we have:

$$T(n) = T\left(\frac{3}{4}n\right) + O(n) = O(n)$$

In fact we can spend $O(n) + T(0.2n)$ time to find a pivot, then

$$T(n) = T\left(\frac{3}{4}n\right) + T(0.2n) + O(n) = O(n)$$.
How to get a good pivot - try random one?

Pick random element $r$ of $A$.

What's $\Pr(r \text{ is a good pivot})$?

Think of sorted $A$:

- smallest $\frac{1}{4}$
- median $\frac{1}{2}$
- largest $\frac{3}{4}$

Exactly $\frac{1}{2}$ good pivots.

$\Pr(r \text{ is a good pivot}) = \frac{n/2}{n} = \frac{1}{2}$

So with prob $\frac{1}{2}$ get a good pivot.

Can check if so in $O(n)$ time.

If not repeat & try a new random element.

In expectation, try twice.

$\Rightarrow O(n)$ expected run time.

But $O(n^2)$ worst case running time.
Recall, aiming for recurrence:

\[ T(n) = T(0.75n) + T(0.2n) + O(n) \]

which solves to \( T(n) = O(n) \).

So we can spend \( T(0.2n) + O(n) \) time to find a good pivot.

- Can choose a subset \( S \) of size \( 0.2n \) and find \( \text{median}(S) \) as the pivot.

How to choose \( S \)?

**Easy:** Let \( S = [a_1, \ldots, a_{0.5}] = 1^{st} \frac{1}{5} \text{ elements of } A \).

But what if these are the \( \frac{n}{5} \) smallest (or largest) of \( A \).

Then: \( p = \text{median}(S) = \frac{10}{10^{th}} \text{smallest of } A \)

which is a bad pivot \( (|A_p| = 0.9n) \)

**Hard:** Choose \( \frac{N}{5} \) random elements.

This works with high probability but hard to analyze, and is randomized.
Want a subset $S$ that represents $A$.
For each $x \in S$, want $x$ to "represent":
- a few elements of $A$ which are $\geq x$
- a few which are $\leq x$

How to achieve this?

Break $A$ into $\frac{N}{5}$ groups of $5$ elements each & then choose a "representative" for each group.

How?

Sort each group & use its median $m_i$.
Why? it’ll have $\geq 2$ that are $\leq m_i$ & $\geq 2$ that are $\geq m_i$.

Takes $O(1)$ time to sort $5$ elements
& thus $O(n)$ time to find these $\frac{N}{5}$ medians $\{m_1, m_2, ..., m_{\frac{N}{5}}\}$.

Then find $p = \text{median}(m_1, m_2, ..., m_{\frac{N}{5}})$.
That gives the pivot.
FastSelect(A,k):

1) Break A into \( \frac{n}{5} \) groups of 5 elements each.
   (assume \( n \) is a power of 5.)
   Call these groups \( G_1, G_2, \ldots, G_{\frac{n}{5}} \).

2) For \( i = 1 \rightarrow \frac{n}{5} \):
   a. Sort \( G_i \)
   b. Let \( m_i = \text{median}(G_i) \)

3) Let \( S = \{ m_1, m_2, \ldots, m_{\frac{n}{5}} \} \).

4) Let \( p = \text{FastSelect}(S, \frac{n}{10}) \). (Hence \( p \) is the median of \( S \)).


6) If \( k \leq |A < p| \) then
   \[ \text{return}(\text{FastSelect}(A < p, k)) \]
   If \( |A < p| < k \leq |A < p| + |A = p| \) then
   \[ \text{return}(p) \]
   If \( |A < p| + |A = p| < k \) then
   \[ \text{return}(\text{FastSelect}(A > p, k-|A < p| - |A = p|)) \]
Claim: P is a good pivot.

Therefore the running time satisfies:

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

\[ \uparrow \text{step 4} \uparrow \text{step 6} \]

Since \[ \frac{1}{5} + \frac{3}{4} = 0.95 < 1 \]

Thus \[ T(n) = O(n) \].

Proof of claim:

Sort \( G_1, \ldots, G_{\frac{n}{5}} \) by their medians so that:

\[ m_1 \leq m_2 \leq \ldots \leq m_{\frac{n}{10}} \leq \ldots \leq m_{\frac{n}{5}} \]

Note \( p = \frac{m_{\frac{n}{10}}}{m_{\frac{n}{5}}} \).
Here's the Picture:

Sort by medians: \( m_1 \leq m_2 \leq \ldots \leq m_{n/10} \leq m_{n/5} \)

\& Sort within each group:

\[
G_i = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ \vdots \end{pmatrix}
\]

so: \( a_1 \leq a_2 \leq a_3 \leq a_4 \leq a_5 \leq m_i \)

Then all of these elements are \( \leq \sigma_i \cdot m_0 \)

\& there are \( \frac{1}{10} \times 3 = \frac{3n}{10} \) such elements.

Note, \( A_{>p} \) excludes these so:

\[
1A_{>p}1 \leq n - \frac{3n}{10} = \frac{7n}{10} < \frac{3}{4} n.
\]

\& similarly for \( A_{>p} \) using