Median:

Given an unsorted list $A=[a_1,\dots,a_n]$ of n numbers

Find the median
(for odd n so $n=2l+1$ then we want to find the $(l+1)^{st}$ smallest)

More generally, given k where $1\le k\le n$
find the kth smallest in A.

Easy algorithm: sort A in $O(n\log n)$ time
then output the kth element in the sorted list.

Our goal: $O(n)$ time algorithm.

Basic idea is like QuickSort:
- Choose a good pivot p (HOW?)
- Partition A based on p
  & recurse on appropriate sublist.

Select(A, k):

1) Choose a pivot p ← this is the challenging step to do
2) Partition A into $A_{<p}$, $A_{=p}$, $A_{>p}$
3) - If $k \le |A_{<p}|$ then
   return(Select($A_{<k}$, k))
   - If $|A_{<p}| < k \le |A_{<p}| + |A_{=p}|$ then
     output p
   - If $k > |A_{<p}| + |A_{=p}|$ then
     return($A_{>p}$, $k - |A_{<p}| - |A_{=p}|$)

Example: $A = [5, 2, 20, 17, 11, 13, 8, 9, 11]$

Say $p = 11$, so: $A_{<p} = [5, 2, 8, 9]$, $A_{=p} = [11, 11]$, $A_{>p} = [20, 17, 13]$
size 4 size 2 size 3

if $k \le 4$, find k^{th} smallest in $A_{<p}$

if $k = 5$ or 6, then 11 is k^{th} smallest

if $k > 6$, then find $(k-6)^{th}$ smallest in $A_{>p}$.

How to choose the pivot p?

Aiming for $O(n)$ running time

So we want a recurrence such as:

$$T(n) = T\left(\frac{9}{10}n\right) + O(n)$$

this solves to $T(n) = O(n)$

in fact for any $c < 1$,

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

So if the pivot p guarantees that:

$$|A<p| < cn \ \& \ |A>p| < cn$$

then we are done.

But even if we have 2 subproblems one of size an & one of size bn if $a+b<1$ so that our recurrence is

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

for $a+b<1$ this solves to $T(n) = O(n)$.

We'll end up with $a = \frac{1}{5}$ & $b = \frac{7}{10}$

by spending $T(\frac{n}{5})$ time to choose a pivot

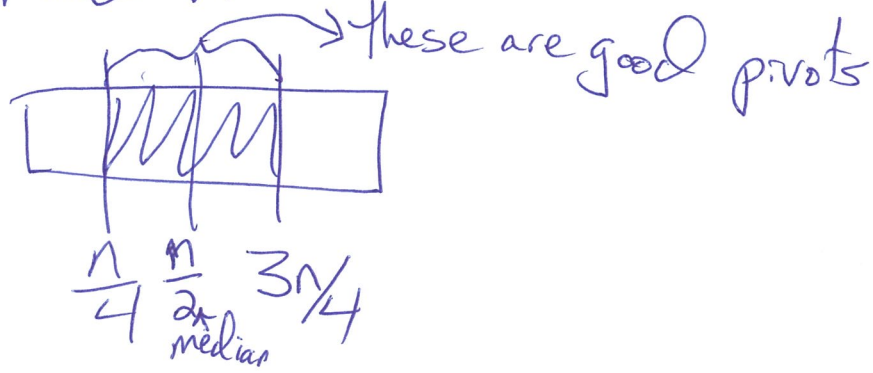
Then this pivot will have: $|A_{<p}| \leq \frac{7}{10} n$

& $|A_{>p}| \leq \frac{7}{10} n$

Let's say a pivot p is <u>good</u> if

$$|A_{<p}| \leq \frac{3}{4} n \quad \& \quad |A_{>p}| \leq \frac{3}{4} n$$

Think of sorted A:



So to find the median we first need a "near-median"

How can we find a good pivot in $O(n)$ time?

Randomized approach:

Choose a random element of A

Check if it's a good pivot

if it is good use it as p

else try again

Probability that random element of A is a good pivot $= \dfrac{\frac{n}{2}}{n} = \dfrac{1}{2}$

In expectation, 2 tries to get a good pivot.

Then expected running time is

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

But this is just expectation, & many times it may be much worse.

So need worst-case running time.

Deterministic approach: [Blum, Floyd, Pratt, Rivest, Tarjan 1973]

Want to choose a subset S of A

We find the median(S) recursively

& use $p = \text{median}(S)$

Say $|S| = \frac{n}{5}$ (the choice of $\frac{1}{5}$ doesn't matter at this point)

Worst case: S is the $\frac{n}{5}$ smallest elements of A.

So $p = \text{median}(S)$ is the $\frac{n}{10}$th smallest of A

thus, $|A_{<p}| \leq \frac{9}{10} n$ & $|A_{>p}| \leq \frac{9}{10} n$

(actually in this case $|A_{<p}| \leq \frac{n}{10}$)

then our running time is

$$T(n) = \underbrace{T\left(\tfrac{n}{5}\right)}_{\text{to find median}(S)} + \underbrace{T\left(\tfrac{9}{10} n\right)}_{\text{if we run on } A>p} + O(n)$$

But $\frac{1}{5} + \frac{9}{10} > 1$ so this doesn't solve to $O(n)$!

Need a more clever choice of S.

For each $x \in S$, want that at least a few elements of A are $\geq x$ & a few are $\leq x$.

So break A into groups of size 5 (assume n is a power of 5)

Let $G_1, G_2, \ldots, G_{\frac{n}{5}}$ be groups of 5 elements each.

Since each group has $O(1)$ elements we can sort a group in $O(1)$ time, and sort all groups in $O(n)$ time.

Let m_i be the median of G_i

~~Let~~ $S = \{m_1, m_2, \ldots, m_{\frac{n}{5}}\}$

This is our S &

let $p = \text{median}(S)$.

We'll see that this gives a good pivot.

FastSelect(A, k):

input: unsorted $A = [a_1, \ldots, a_n]$ where n is a power of 5
& integer k where $1 \le k \le n$

output: k^{th} smallest of A

1) Break A into $\frac{n}{5}$ groups of 5 elements each.
Call these groups $G_1, G_2, \ldots, G_{\frac{n}{5}}$

2) For $i = 1 \to \frac{n}{5}$, sort G_i

3) Let $m_i = \text{Median}(G_i)$
Let $S = \{m_1, m_2, \ldots, m_{\frac{n}{5}}\}$

4) $p = \text{FastSelect}(S, \frac{n}{10})$ (so p is the Median(S))

~~5) If $k \le |A_{<p}|$~~

5) Partition A into $A_{<p}, A_{=p}, A_{>p}$

6) If $k \le |A_{<p}|$ then return(FastSelect$(A_{<p}, k)$)
If $|A_{<p}| < k \le |A_{<p}| + |A_{=p}|$ then return(p)
If $k > |A_{<p}| + |A_{=p}|$ then
return$(A_{>p}, k - |A_{<p}| - |A_{=p}|)$

<u>Claim:</u> p is a good pivot

In particular,

$\geq \frac{3n}{10}$ elements of A are $\leq p$ so $|A_{>p}| \leq \frac{7n}{10}$

& $\geq \frac{3n}{10}$ elements are $\geq p$ so $|A_{<p}| \leq \frac{7n}{10}$

From the claim, the running time is thus:

$$T(n) \leq \underset{\text{step (4)}}{T\left(\frac{n}{5}\right)} + \underset{\text{step (6)}}{T\left(\frac{7n}{10}\right)} + O(n)$$

Since $\frac{1}{5} + \frac{7}{10} = \frac{9}{10} < 1$

then $T(n) = O(n)$.

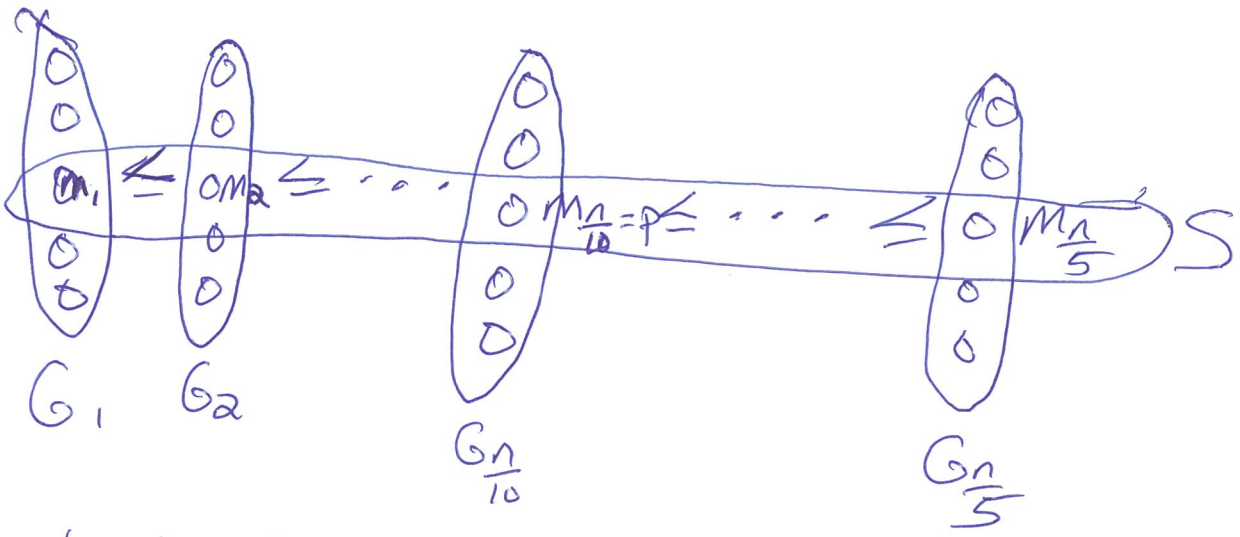
Proof of Claim:

Sort the groups by their medians so that:

$$m_1 \le m_2 \le \cdots \le m_{\frac{n}{5}}$$

Then $p = m_{\frac{n}{10}}$

Here's the picture:



which elements <u>of S</u> are guaranteed to be $\le p$?

$m_1, m_2, \ldots, m_{\frac{n}{10}}$ are $\le m_{\frac{n}{10}} = p$.

& for each of these, 3 elements in its group are $\le$ it.

Hence, $\ge (3)\left(\frac{n}{10}\right)$ are $\le p$ ✓

Similarly, $\ge \frac{3n}{10}$ are $\ge p$ ✓ $\square$