Hw 2 due today

Hw 3 → Practice exam

Quiz 1 next Thursday

Review sessions Tu, W
1-3, 5-7

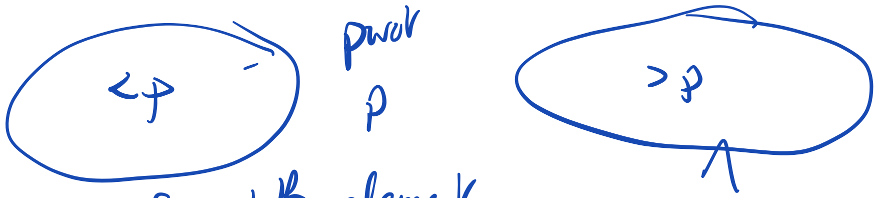
---

1. QuickSort takes $O(n \log n)$ time if we have good pivots
2. If we could find the median, we have a good (the best) pivot
3. Median is not self-reducible, but Selection is: You can break selection into smaller problems of the same type, and median is a special case
4. Selection is $O(n)$ if we get good pivots each time.

sorted order

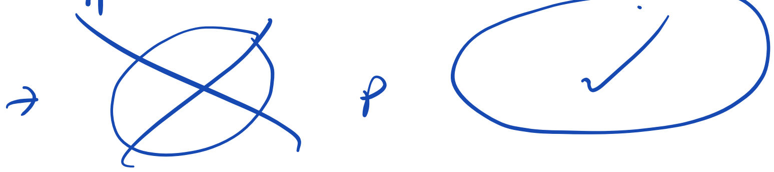
$[A] = [\quad | \underbrace{\quad \text{Good} \quad}_{} | \quad]_n$

$\frac{1}{4}n \qquad \frac{3n}{4}$

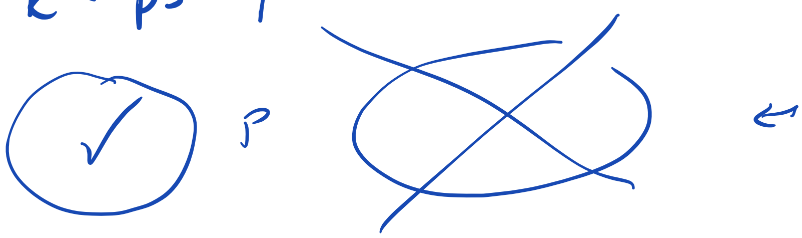


Looking for k^{th} element

if $k >$ p's position



If $k <$ p's posn



$k =$ p's posn, output p.

Want to eliminate $\geq \frac{1}{4}$ of remaining items at each step

because then $T(n) \leq T\left(\frac{3n}{4}\right) + O(n)$

$$\Rightarrow T(n) \leq O(n)$$

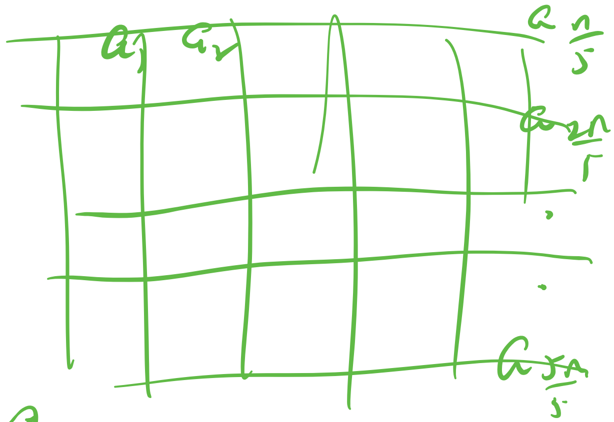
Finding a "good" pivot:

input $a_1 \ldots$ — — a_n

Selection takes $T(n)$ time

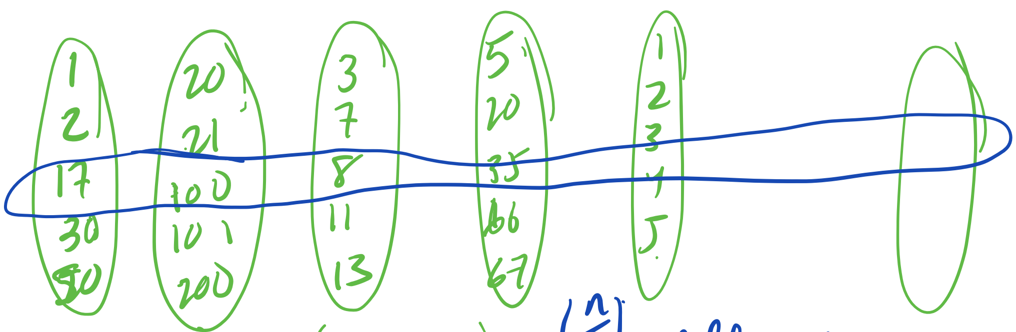
$5 \to \frac{n}{5}$ arrays $O(n)$

1. Rearrange ...



$\frac{n}{5}$ columns

2. Sort each of the $\frac{n}{5}$ colums. $O(n)$ time



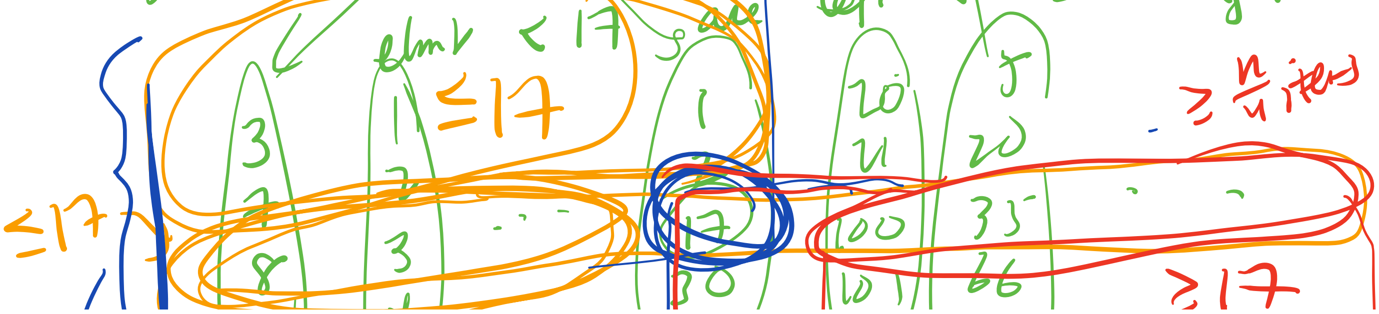
Find the median of the $(\frac{n}{5})$ middle elements

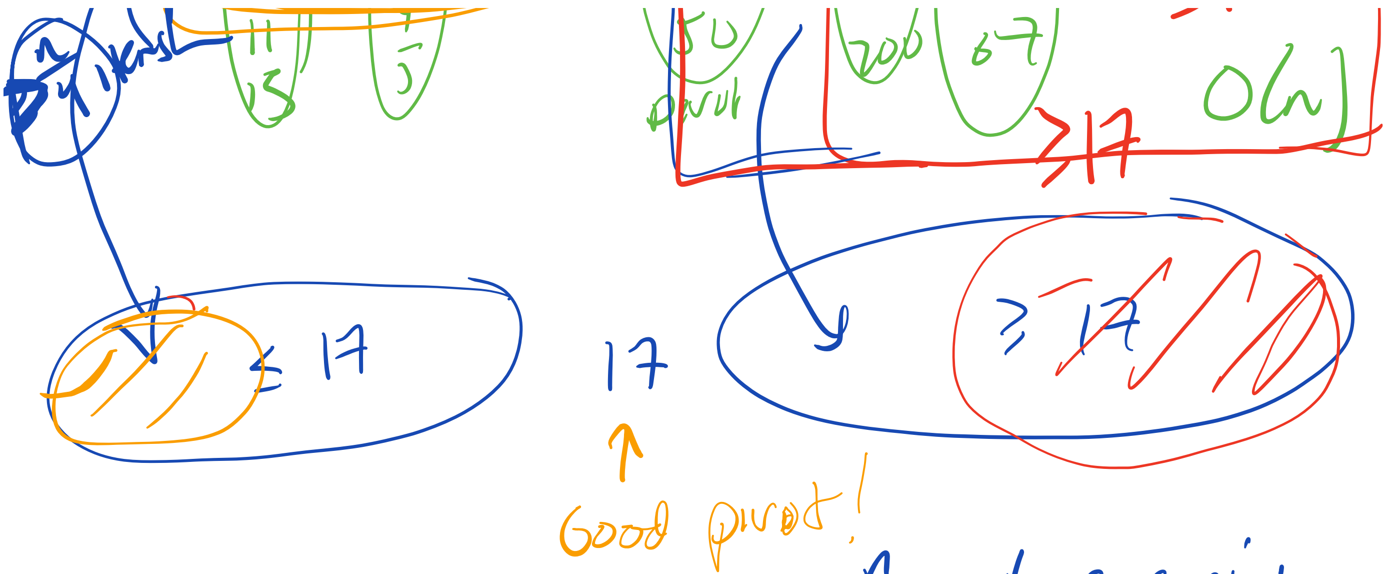
selection

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

Maybe 17 is median

3. Reorder columns so that ones with middle elmt < 17 are left + ≥ 17 right





Now I eliminate $\geq \frac{n}{4}$ of remaining items

$$\Rightarrow \quad T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{3n}{4}\right) + O(n)$$

recursive

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

Version of Master Thm

If $T(n) = T(c_1 n) + T(c_2 n) + \dots + T(c_k n) + O(n^d)$

And $c_1 + c_2 + \dots + c_k < 1$

then $T(n) = \begin{cases} O(n^d) & \text{if } d \geq 1 \\ O(\log n) & \text{if } d = 0 \end{cases}$

Dynamic Programming

Fibonacci sequence

$a_0 = 1$

$a_1 = 1$

$a_n = a_{n-1} + a_{n-2}$

$\underbrace{1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad 21 \quad \_\_ \quad \_\_}_{100}$

What is a_{100}?

$O(n)$ time

a_{100}

$a_{99} + a_{98}$

$a_{98} \quad a_{97} \quad a_{97} \quad a_{96}$

2^n leaves

n levels
we double each time