ORDER STATISTICS - MEDIAN FINDING, RANK SELECTION

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

↳ easy $O(n)$ if $k = O(1)$ or $n - O(1)$.
(median is hardest)

We will assume distinct elements.

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

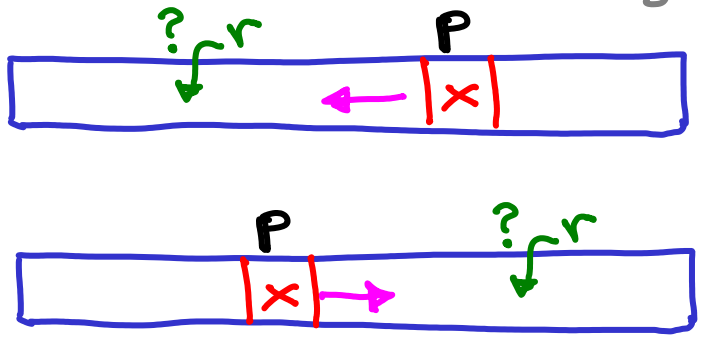
1973

Let this run in time T(n)

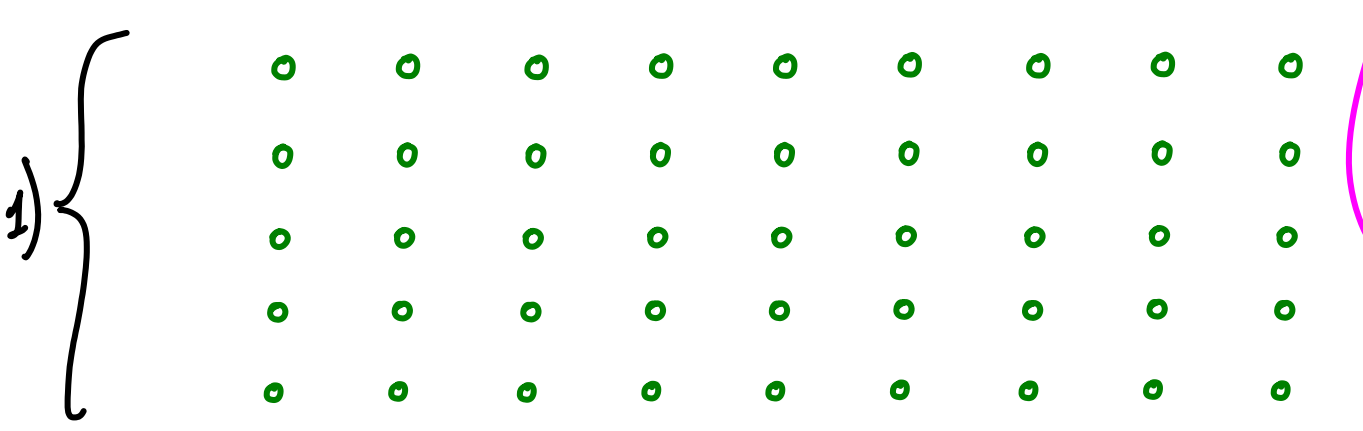
we are looking for the element w/ rank r

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 <5
2) Find median in each group // brute force.
3) Recursively find x = median-of-medians
4) Compare all elements to x → 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
   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 = ∞

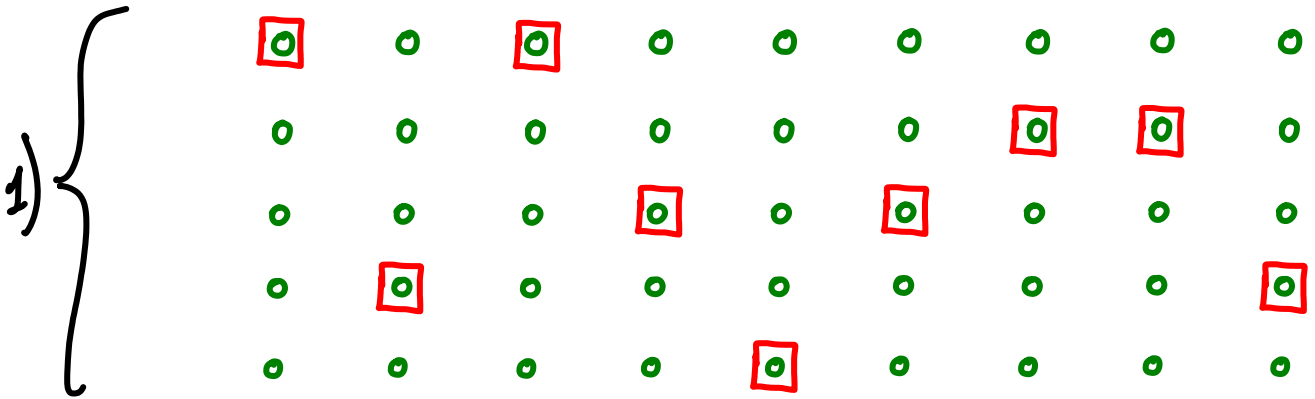
OR

remove MAX & MAX-1

$\Theta(n)$

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

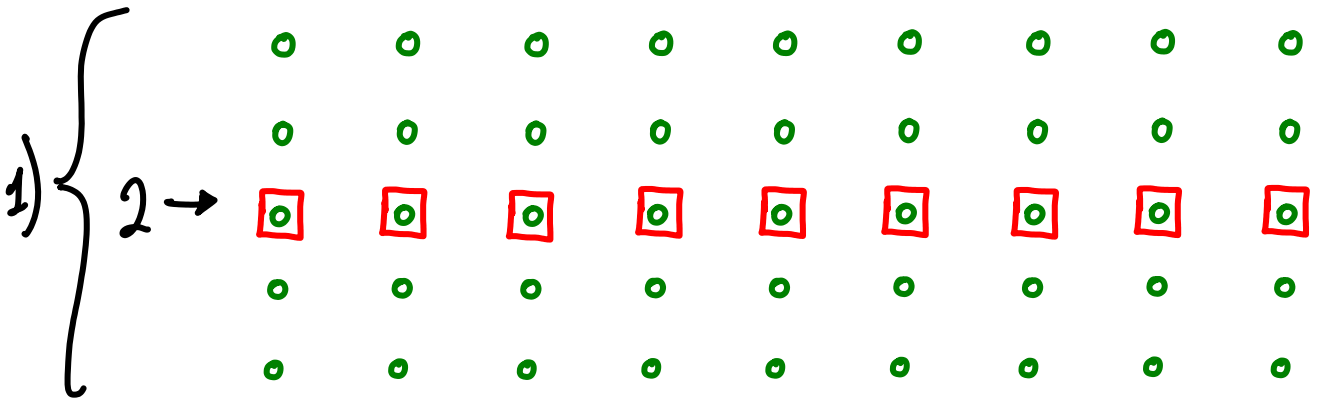
2) Find median in each group $\frac{n}{5} \cdot \Theta(1) = \Theta(n)$



23 7 40 8 12 → 23 7 40 8 **12**

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

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

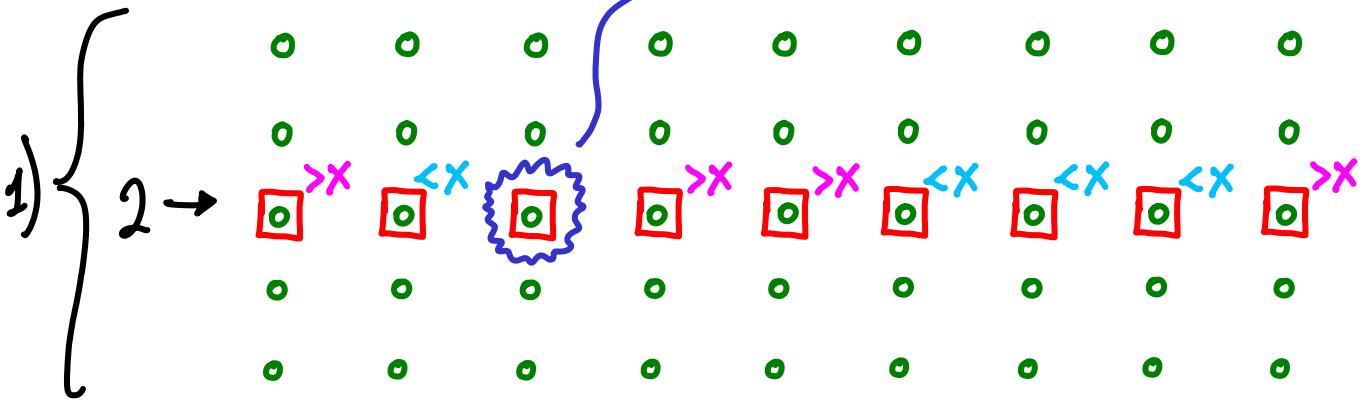


23	23	23	>12
7	7	40	>12
40 →	40 →	12	
8	8	7	<12
12	12	8	<12

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

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

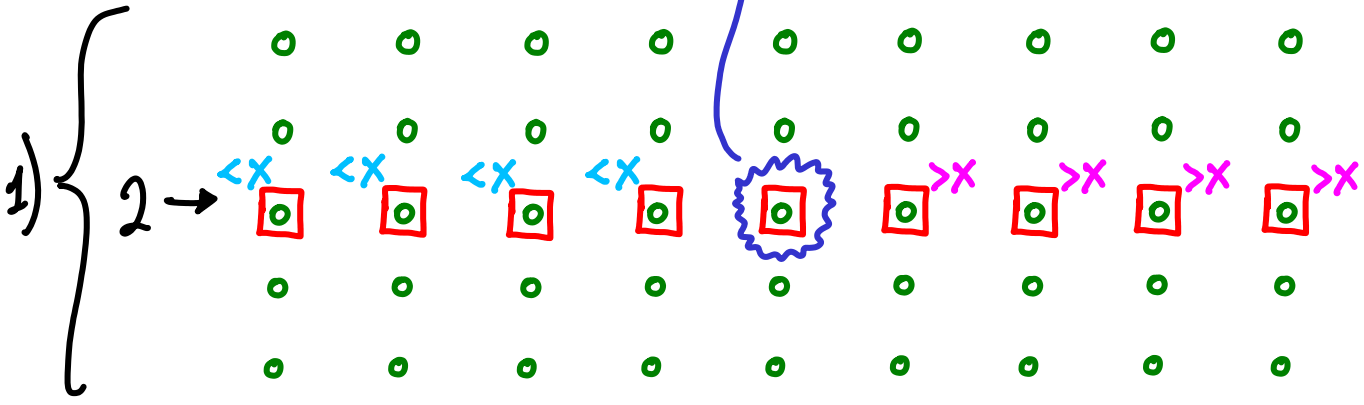
3) Recursively find x = median-of-medians TIME $\longrightarrow$ $T(\frac{n}{5})$



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

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

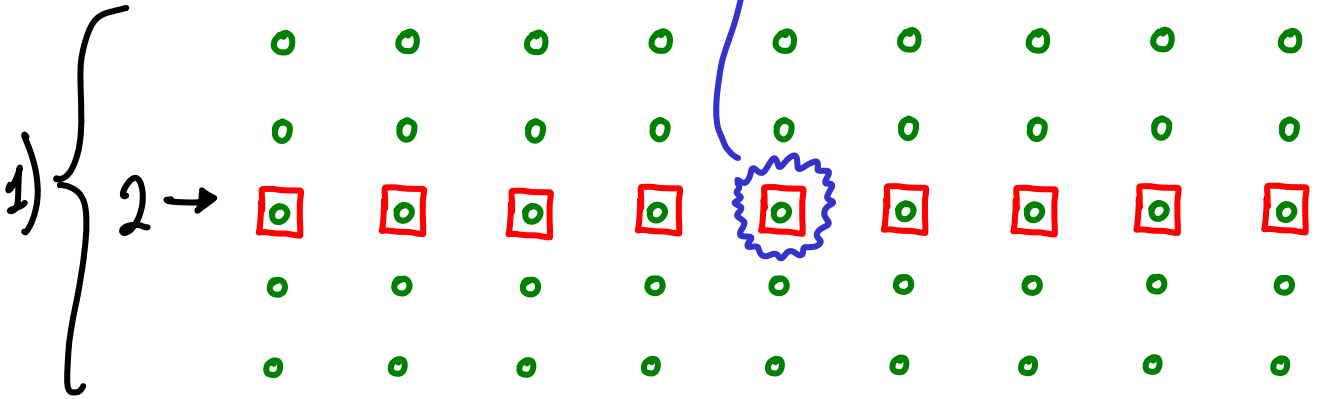
3) Recursively find x = median-of-medians (and re-organize) $T(\frac{n}{5}) + \Theta(n)$



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

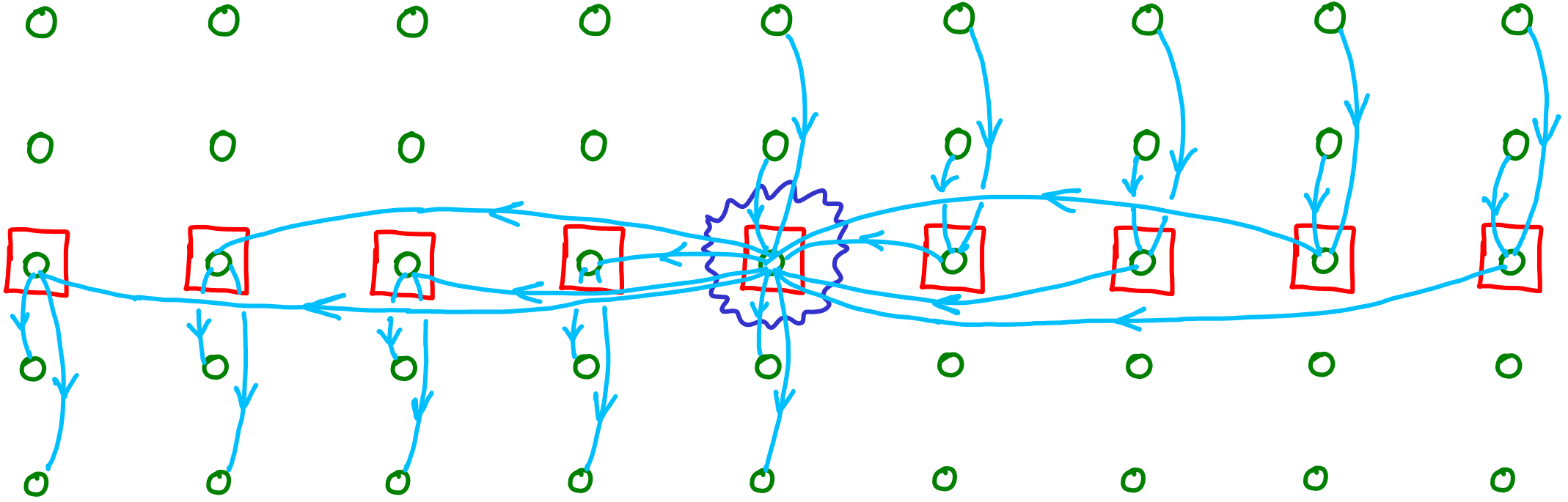
2) Find median in each group (and re-organize) $\frac{n}{5} \cdot \Theta(1) = \Theta(n)$

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



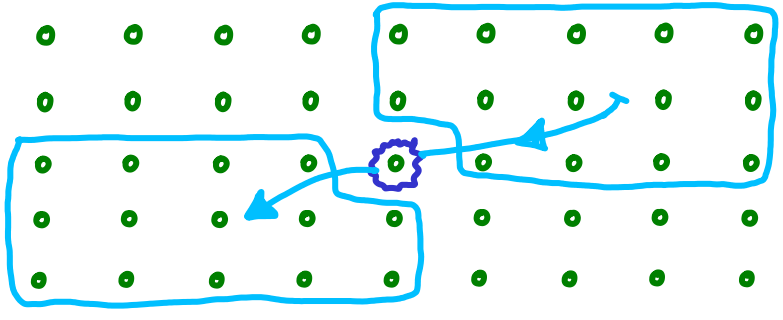
Re-organizing is not part of the algorithm. It's part of the proof.
(although we could afford it)

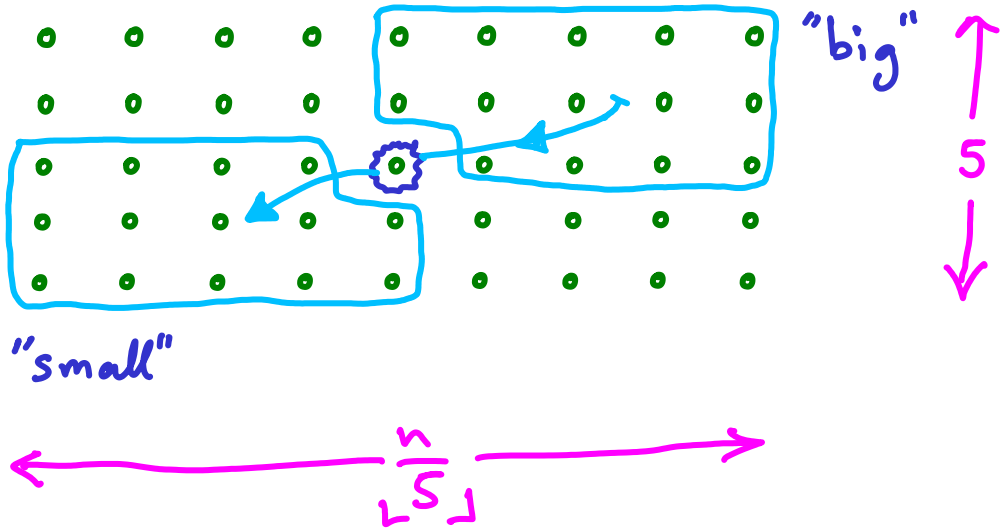
That's the algorithm. Now to find $T(n)$

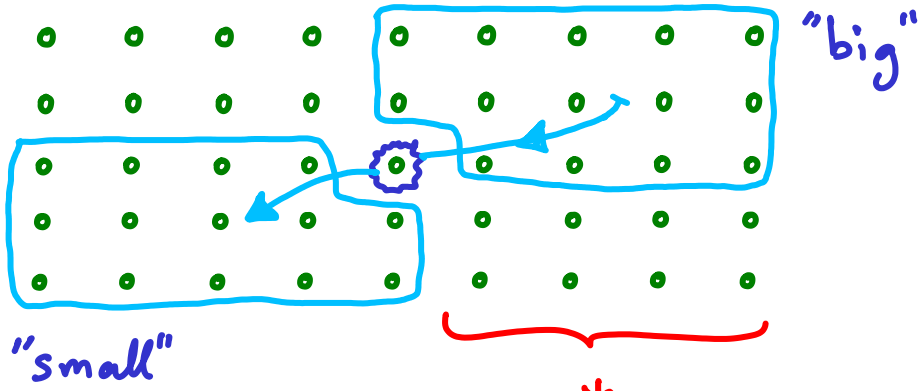


Let 

mean $X > Y$




"big"
5
"small"
n/5

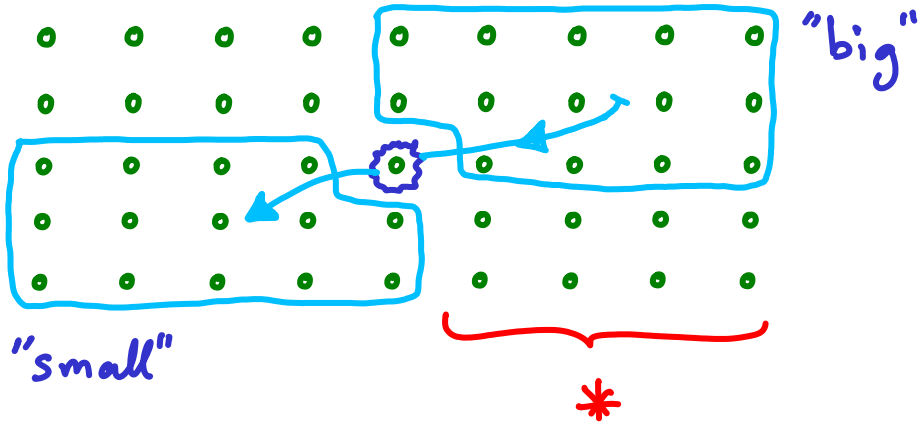


$$\#\text{"big"} = \#\text{"small"} \geqslant 3 \cdot \left\lfloor \frac{n/5}{2} \right\rfloor$$

*

columns containing big elements

big items per column



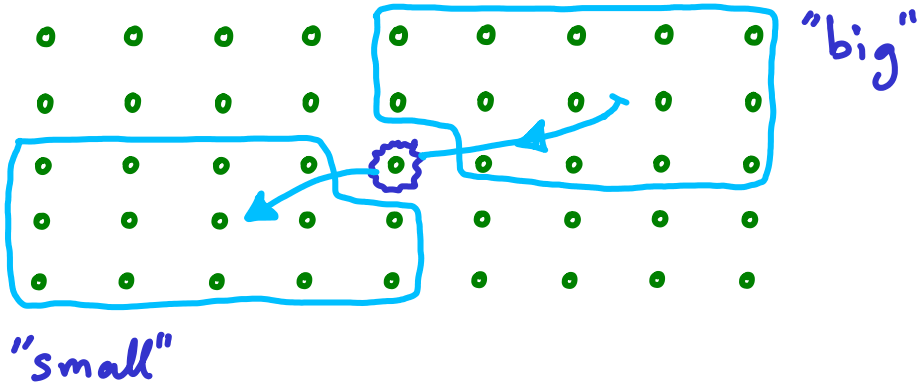
$$\#\text{"big"} = \#\text{"small"} \geq 3 \cdot \frac{\lfloor n/5 \rfloor}{\lfloor 2 \rfloor} \geq 3 \cdot \frac{n}{\lfloor 10 \rfloor}$$

*

$\frac{n}{\lfloor 5 \rfloor} \to \frac{n}{5}$ if we ignore incomplete column

$\frac{n/5}{\lfloor 2 \rfloor} \to \frac{n/5}{2}$ if n: even

$\Theta(n)$ work takes care of this

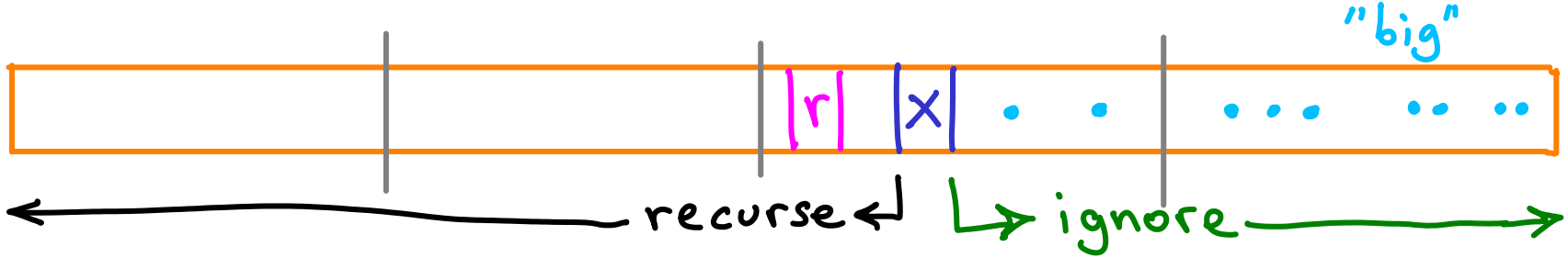


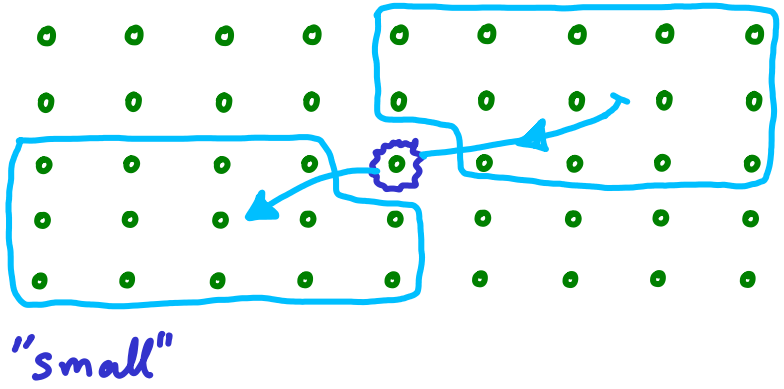
$$\#\text{"big"} = \#\text{"small"} \geq 3 \cdot \frac{\lfloor n/5 \rfloor}{\lfloor 2 \rfloor} \geq 3 \cdot \frac{n}{\lfloor 10 \rfloor}$$

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

if (x) 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"]





$\#\text{"big"} = \#\text{"small"} \geq 3 \cdot \frac{\lfloor n/5 \rfloor}{\lfloor 2 \rfloor} \geq 3 \cdot \frac{n}{\lfloor 10 \rfloor}$

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

if (x) is not at the target rank/index, and we need to search lower (i.e., 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

steps 1&2: split into groups & find medians of 5 & partition

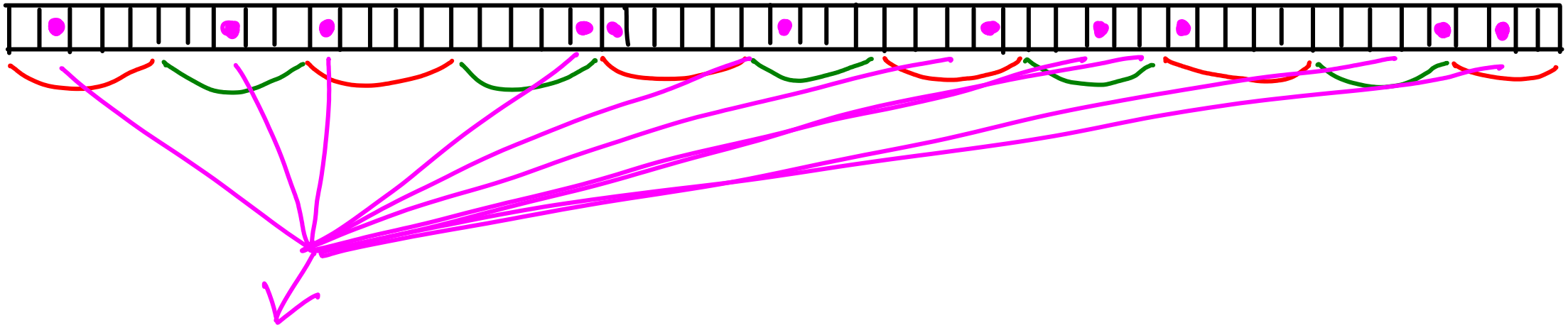
$$T(n) \leq T\left(\frac{n}{5}\right) + T\left(\frac{3n}{4}\right) + \Theta(n)$$

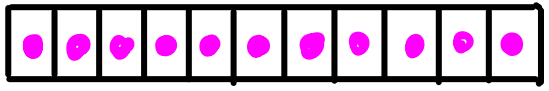
Claim $T(n) \leq c \cdot n$

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

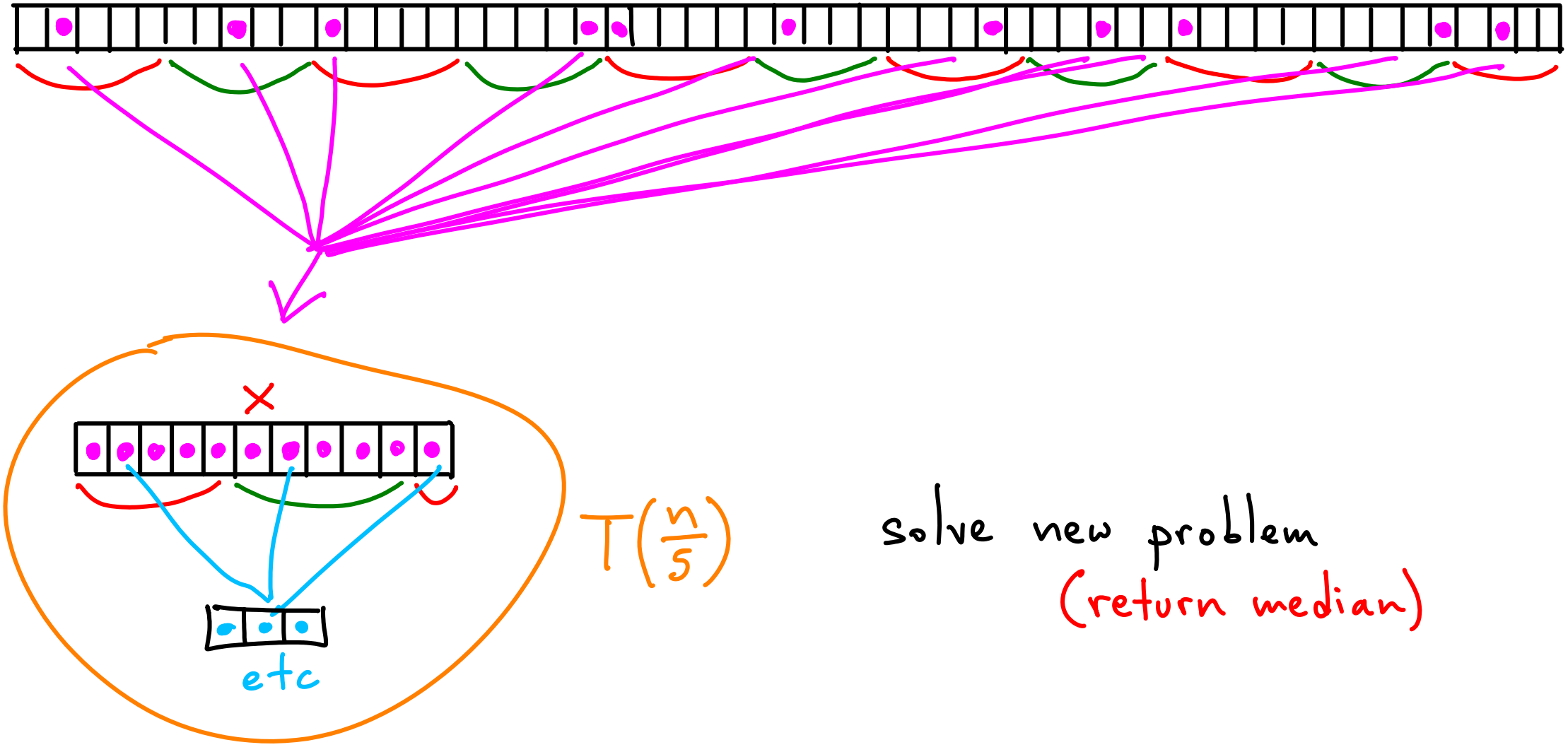
$$= \frac{19}{20} cn + dn = cn - \left(\frac{1}{20} cn - dn\right) \leq cn \text{ if } c > 20d$$

QED

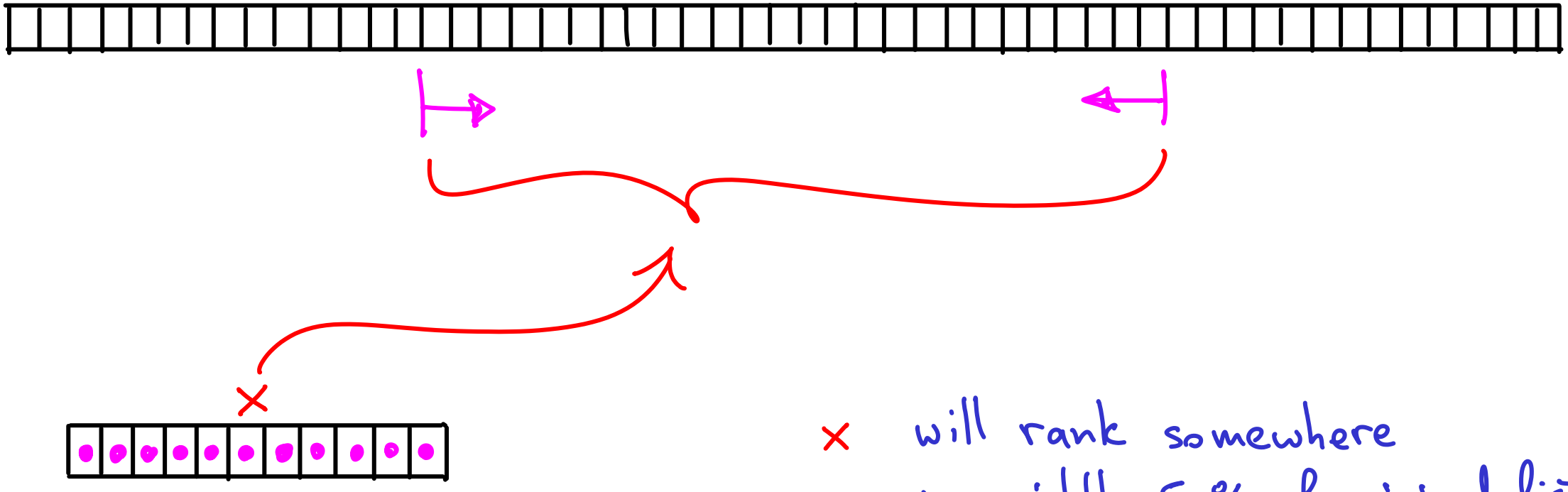




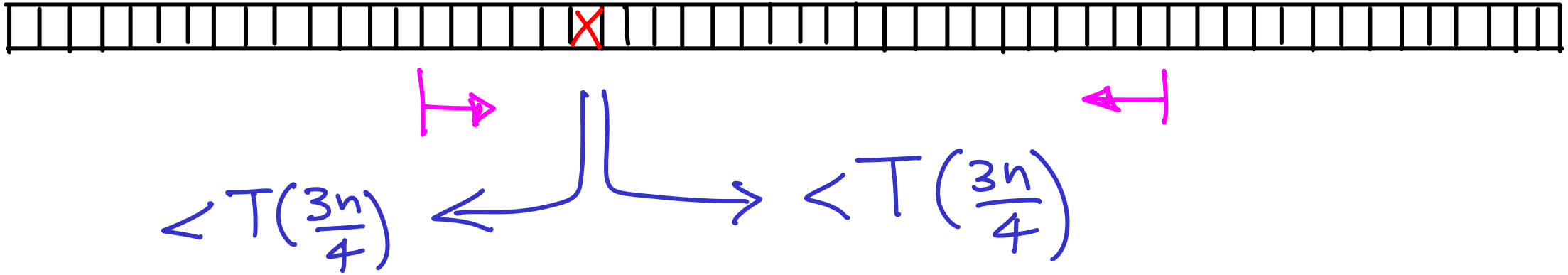
collect medians-of-5

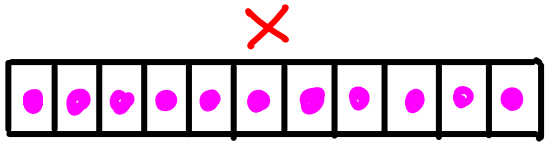


solve new problem
(return median)



× will rank somewhere
in middle 50% of original list





Find rank(x)

if x ≠ target, recurse on $< \frac{3}{4}$ of list

What were they thinking? (my guess)

- Goal: $\Theta(n)$ [$\Omega(n)$ lower bound; $O(n \log n)$ is trivial]
- Exploit ~geometric series: $T(n) = T(\frac{n}{b}) + O(n)$

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

... where $x + y < 1$

↳ spend $T(xn) + O(n)$ time

to make sure that only yn candidates remain