RAM model of computation

Entire input resides in RAM. But bringing from DRAM >> CPU Time.

CPU Time = free

Time Complexity = No. of scans to solve it.

RAM Size = $O(\sqrt{n})$

Algo		Algo A	Algo B	Algo C
No. of scans	$O(n)$	$\frac{\sqrt{n}}{2}$	$O(\log n)$	2 optimal

$n = 2^{2K}$ for some K, all distinct.

$\sqrt{n}/2$ scans

- Compute smallest $\sqrt{n}$ elements in 1 scan.
- Compute next smallest $\sqrt{n}$ elements in 1 scan.
- Keep computing for $\frac{\sqrt{n}}{2}$ times.

$O(\log n)$ scans

→ we find an element with rank $\in \left[\frac{n}{4}, \frac{3n}{4}\right]$ in 1 scan.

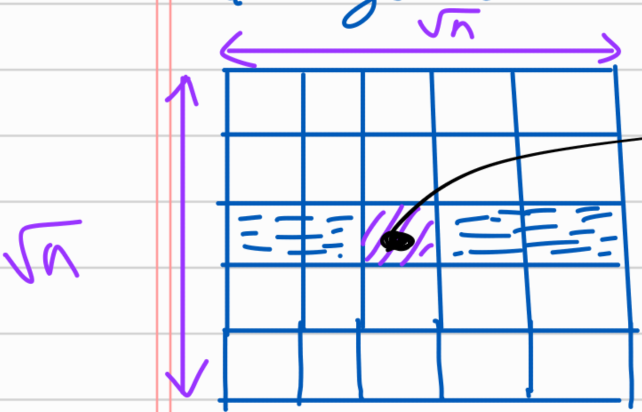
→ find its rank in entire array

→ if rank $< \frac{n}{2}$ ⇒ remove all elements smaller than it.

→ if rank $> \frac{n}{2}$ ⇒ remove all elements greater than it greater than x.

→ find the element in rem array.

→ repeat $O(\log n)$ times to bring down the size to $O(\sqrt{n})$



$x \rightarrow \frac{\sqrt{n}}{2} \times \frac{\sqrt{n}}{2} \Rightarrow \frac{n}{4}$

rank $\in \left[\frac{n}{4}, \frac{3n}{4}\right]$

middle row → row of medians of each $\sqrt{n}$ size array. x = median of medians.

2 Scans Algo

→ We find an element with rank $\in \frac{n}{2} \pm \frac{\sqrt{n}}{2}$

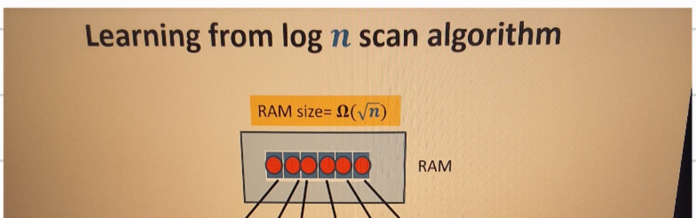
→ Keep largest $\frac{\sqrt{n}}{2}$ numbers less than x

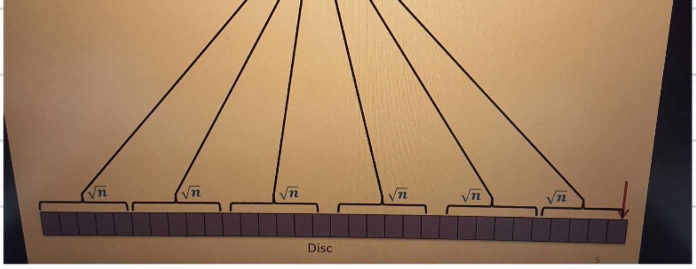
→ Keep smallest $\frac{\sqrt{n}}{2}$ nos. greater than x.

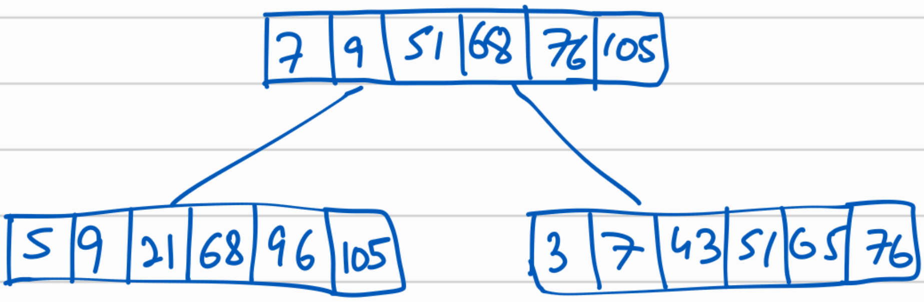
Well separated numbers

The $\sqrt{n}$ numbers we keep in the RAM from each set of $\sqrt{n}$ be well separated.

Any element with rank i → in the root ($\sqrt{n}$ ele) in the sample has rank $i\sqrt{n}$ → in n elements. in the set.







- Sort each contiguous chunk of $\sqrt{n}$ numbers.
- pick alt numbers in the sample.