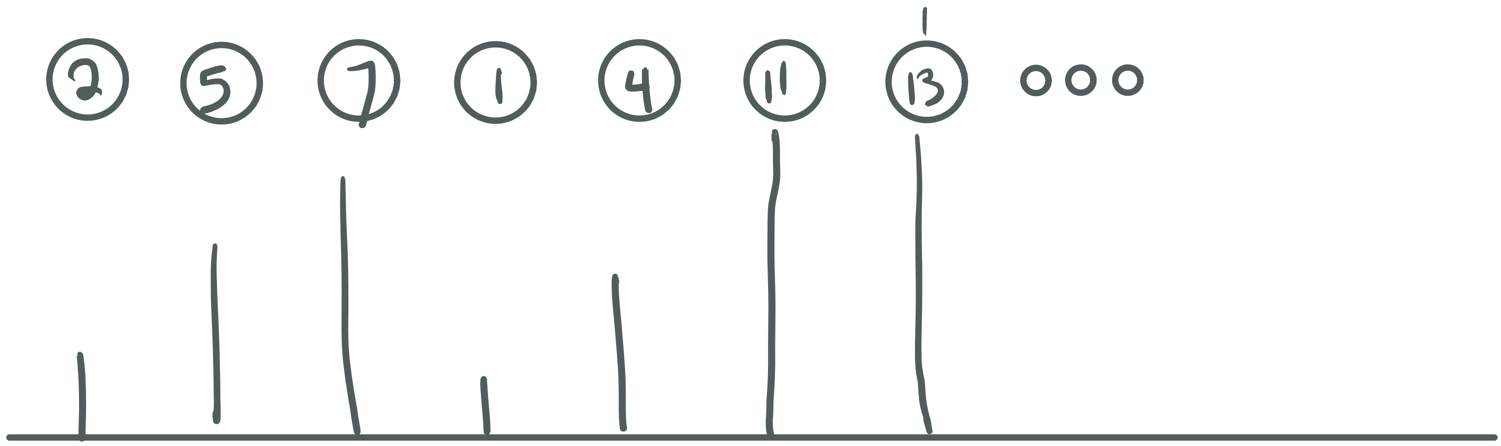
Space efficient quantile selection

Input: stream $a_1, \ldots, a_n \in \mathcal{U}$ where $\mathcal{U}$ has order $<$



e.g. numerical data
names w/ alphabetic order
grades

allowed multiple passes

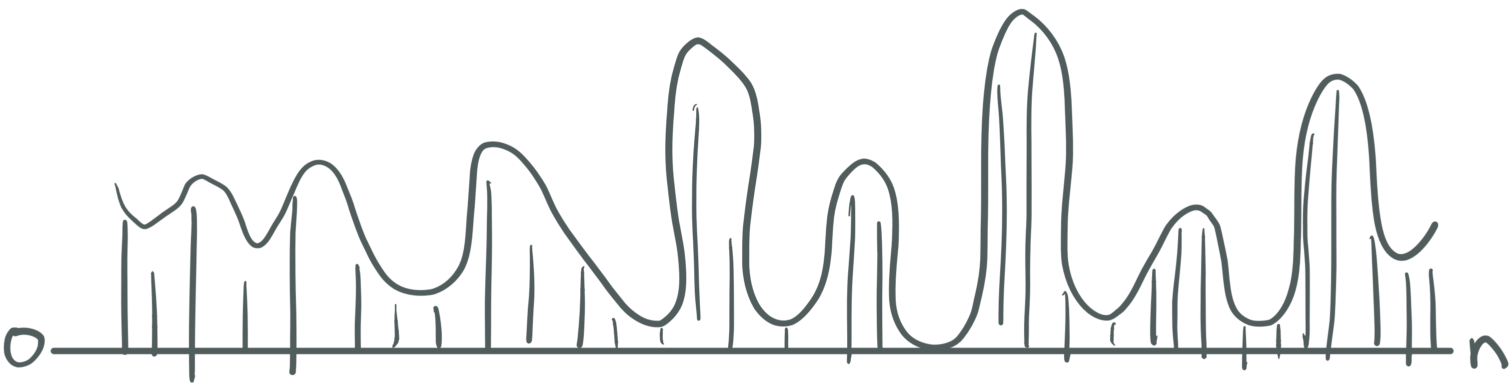
Goal: return the median w/ minimum:

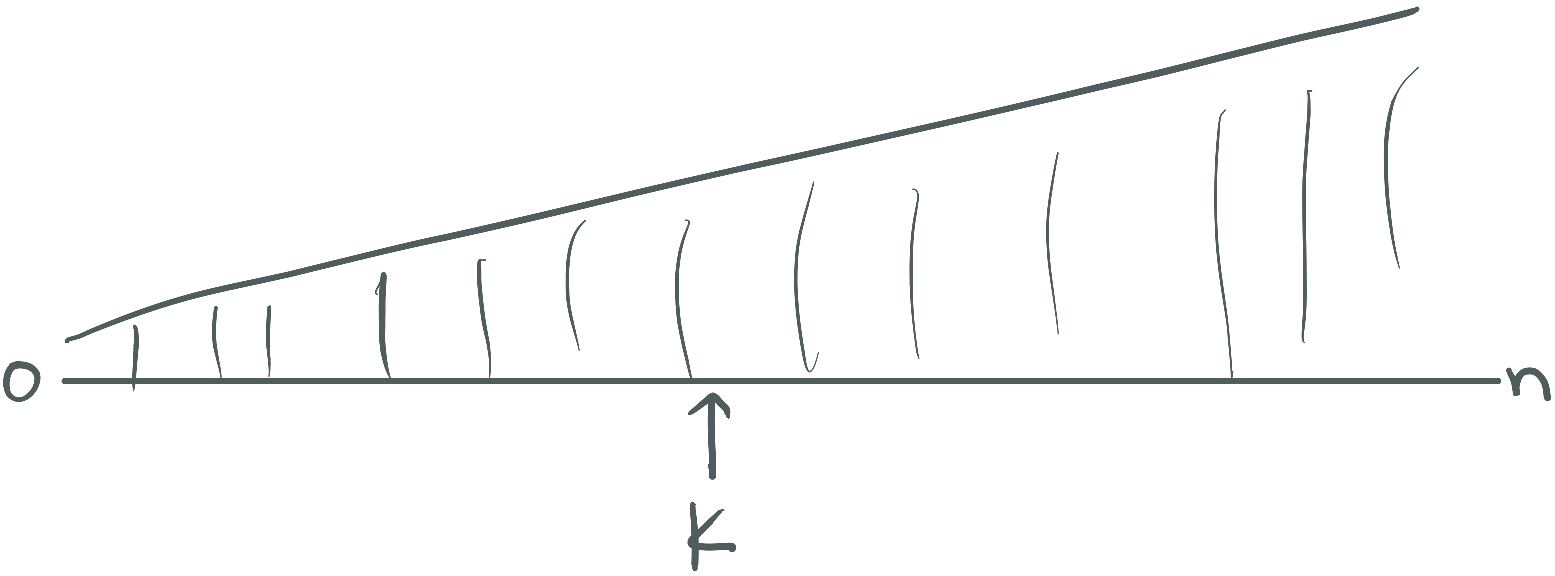
(a) # passes (b) space

more generally: "quantile queries"

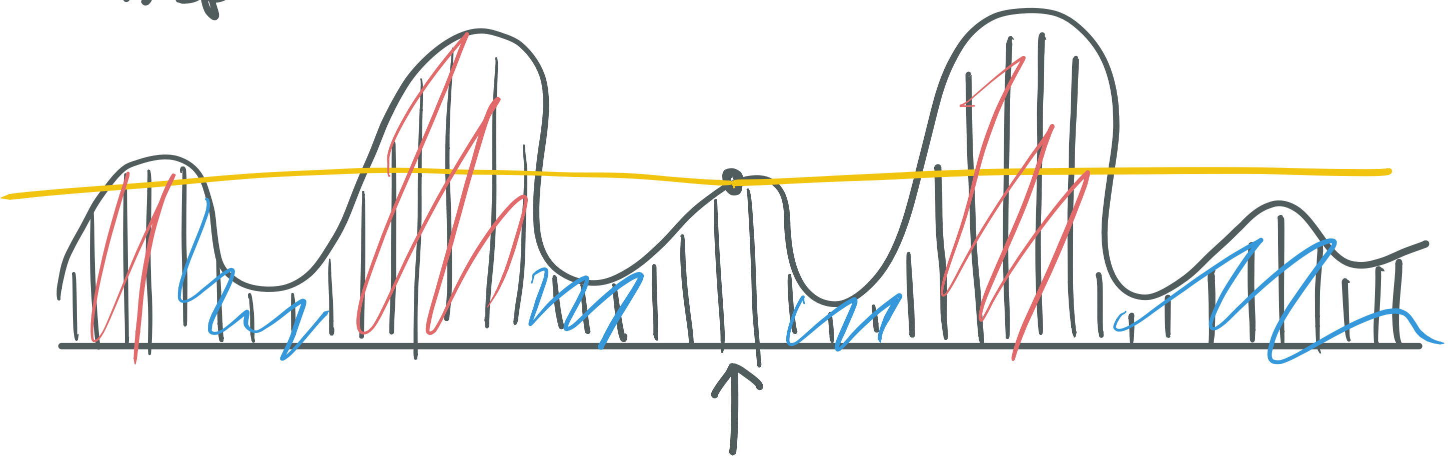
select rank k element
(kth largest)

1 pass: Input





O(1) space

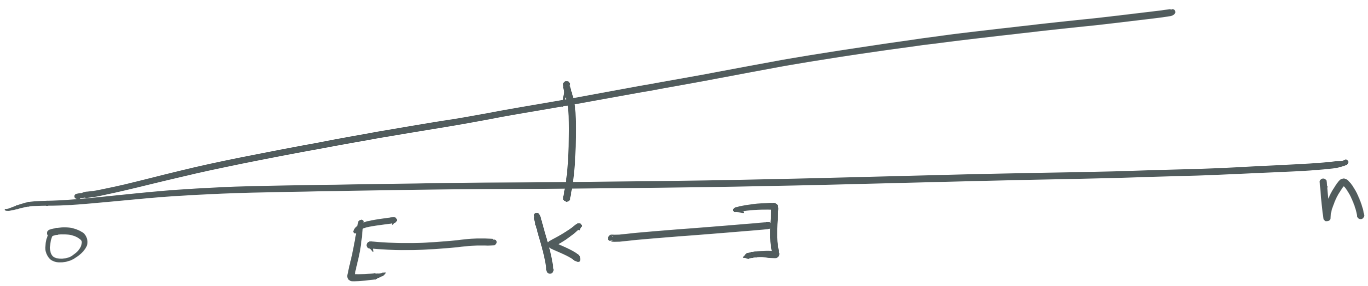


Passes	Space	
1	$O(n)$	sort and select
$O(\log n)$	$O(1)$	quickselect (random pivot)
p	$\tilde{O}(n^{1/p})$	Munro Paterson 1981
2	$\sqrt{n}$	

quantile summaries

Approximations

given rank $k \in [n]$ and param $\varepsilon > 0$,
return element w/ rank $k \pm \varepsilon n$



Sampling:

for median:

sample $\ell = O\left(\frac{\log 1/\delta}{\varepsilon^2}\right)$ elements

return median of sample

for rank $k = \alpha n$

sample ℓ

return rank $\alpha \ell$ out of the sample

Deterministic?

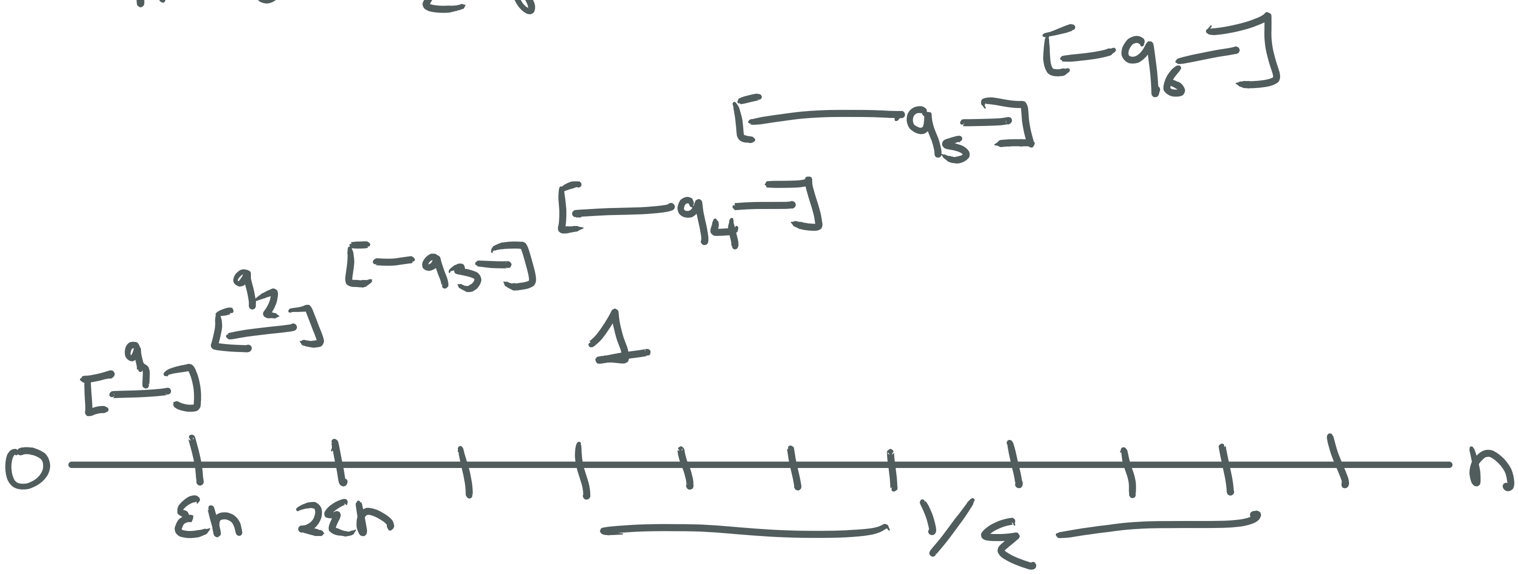
Quantiles

space-efficient
mergable
answer ε-approximate quantile queries

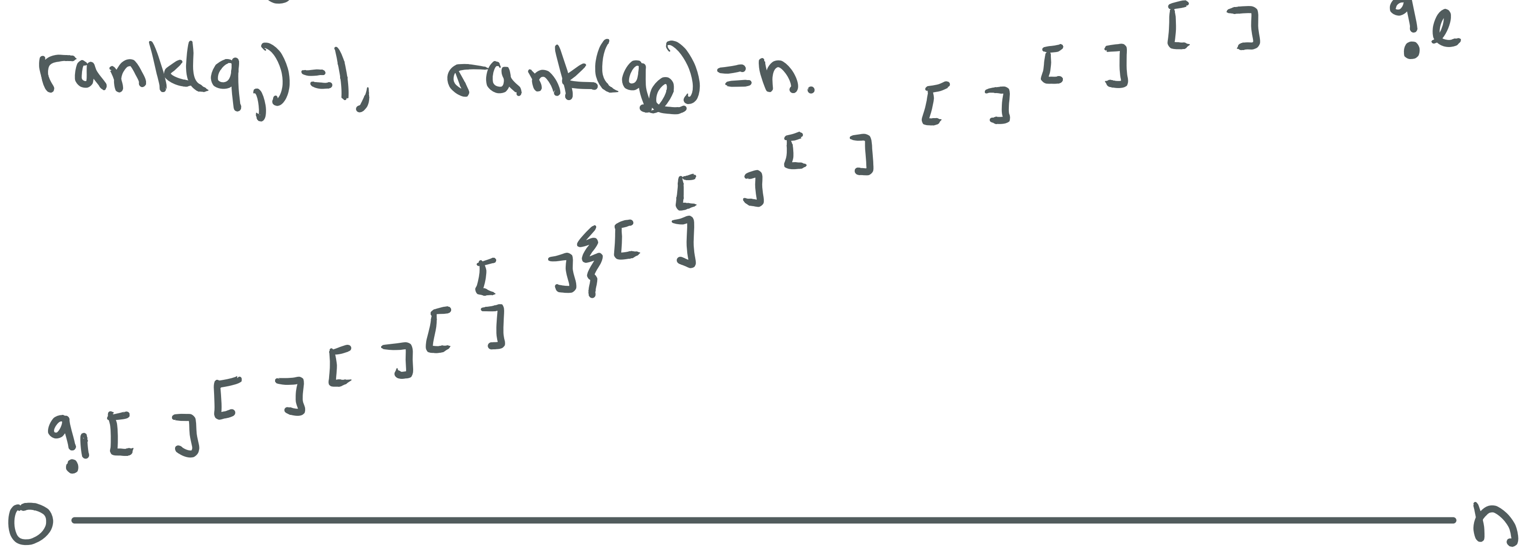
ℓ elements $q_1 < q_2 < \cdots < q_\ell \in S$

along w/ intervals $I(q_i)$ $\quad$ $\mathrm{rank}(q_i) \in I(q_i)$

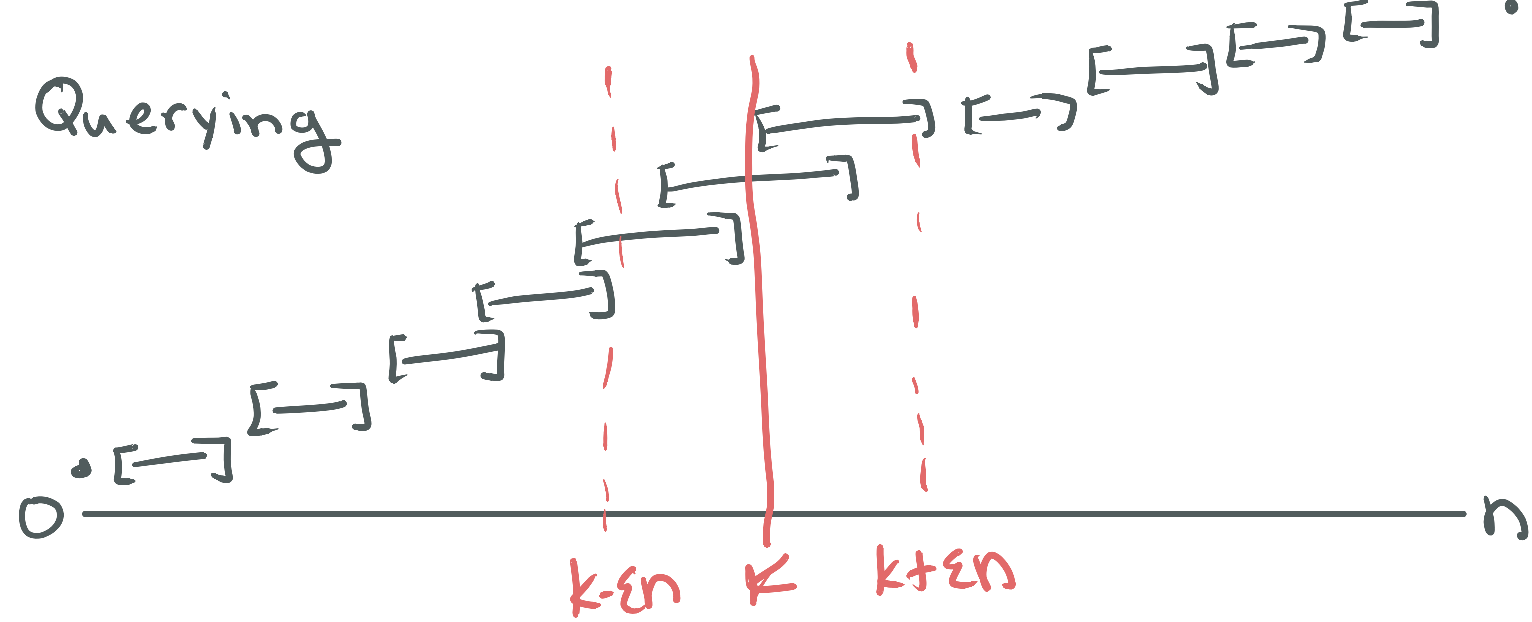
need $\geq \frac{1}{\varepsilon}$ points

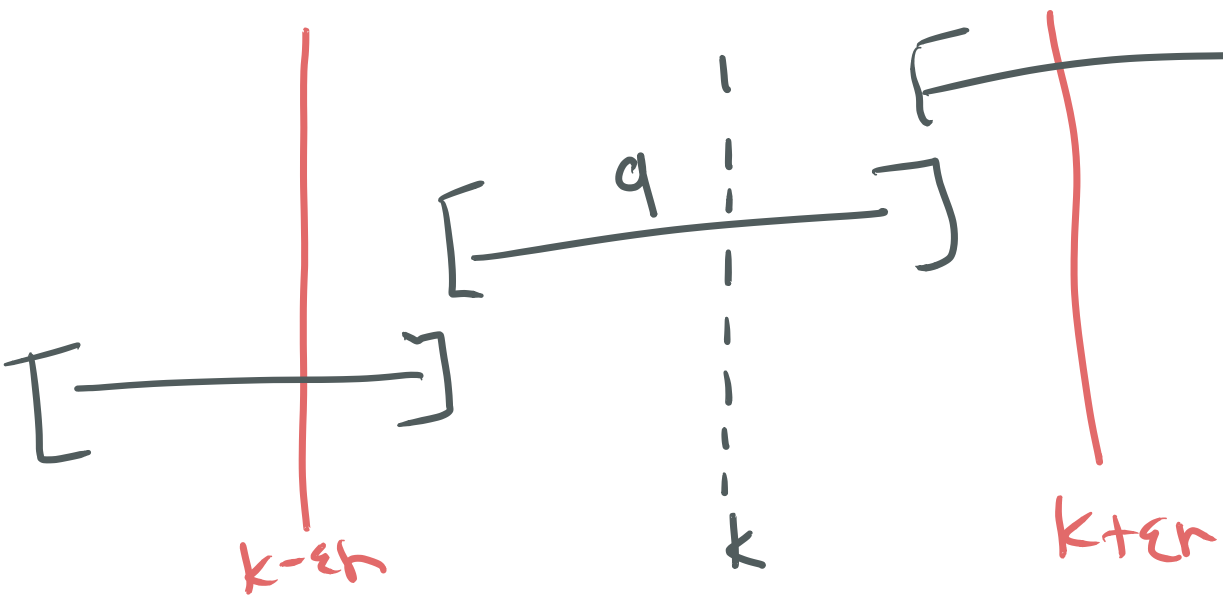


By tracking min/max specially, assume
$\mathrm{rank}(q_1) = 1$, $\mathrm{rank}(q_\ell) = n$.



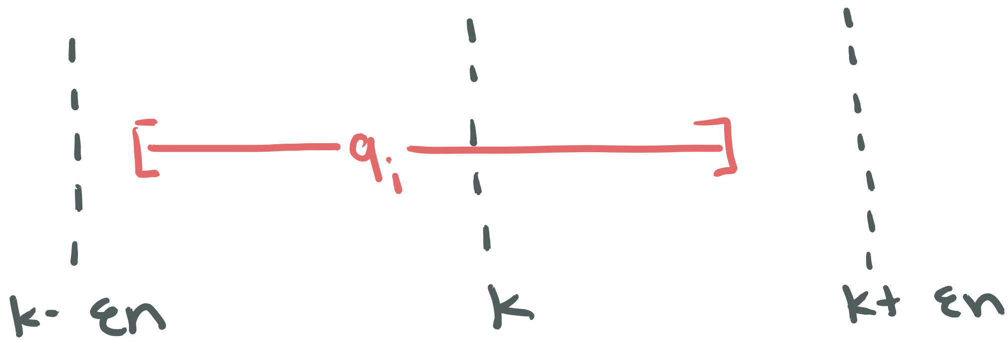
Querying





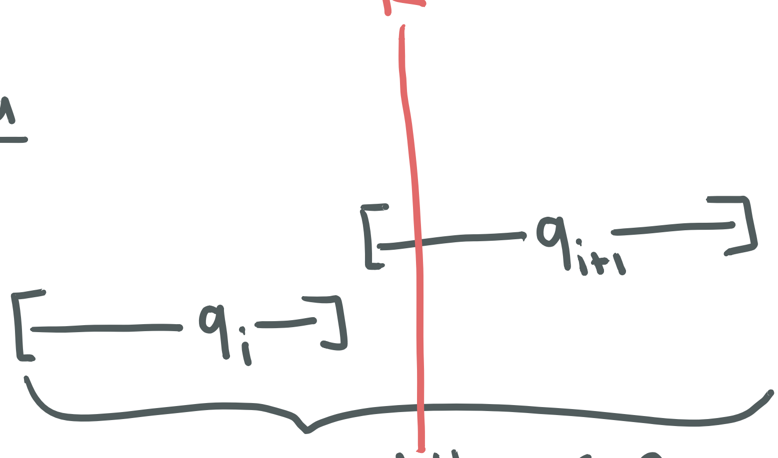
If $I(q_i) \subseteq [k-\varepsilon n, k+\varepsilon n]$ for some q_i,

then return q_j.



how to ensure such q_i exists $\forall k$?

lemma

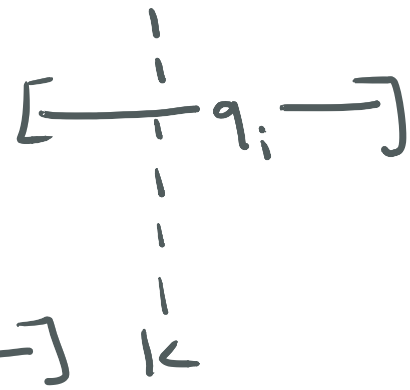


total width $\leq 2\varepsilon n$ $\quad \forall i$

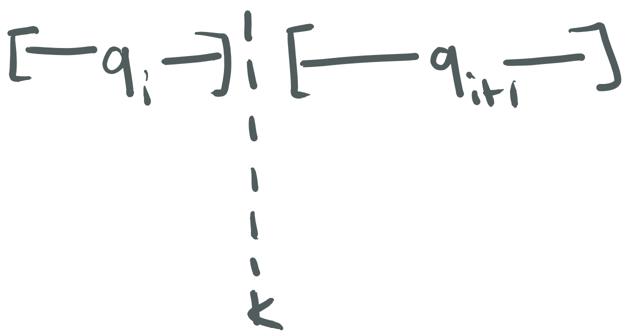
Then every query k contains an interval $I(q_i)$.

Proof two cases:

$k \in I(q_i)$ for some q_i



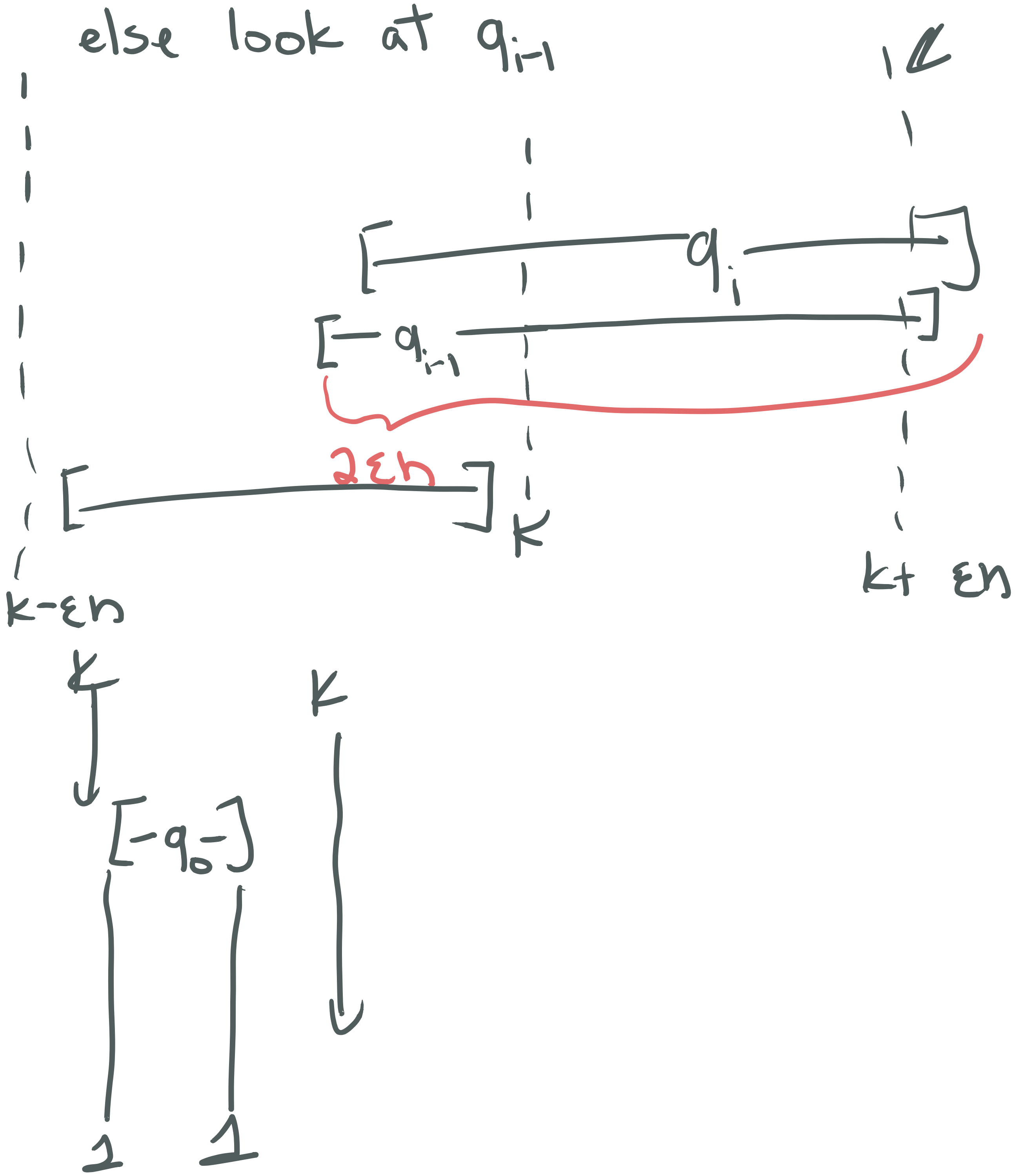
$k \notin I(q_i)\ \forall q_i$



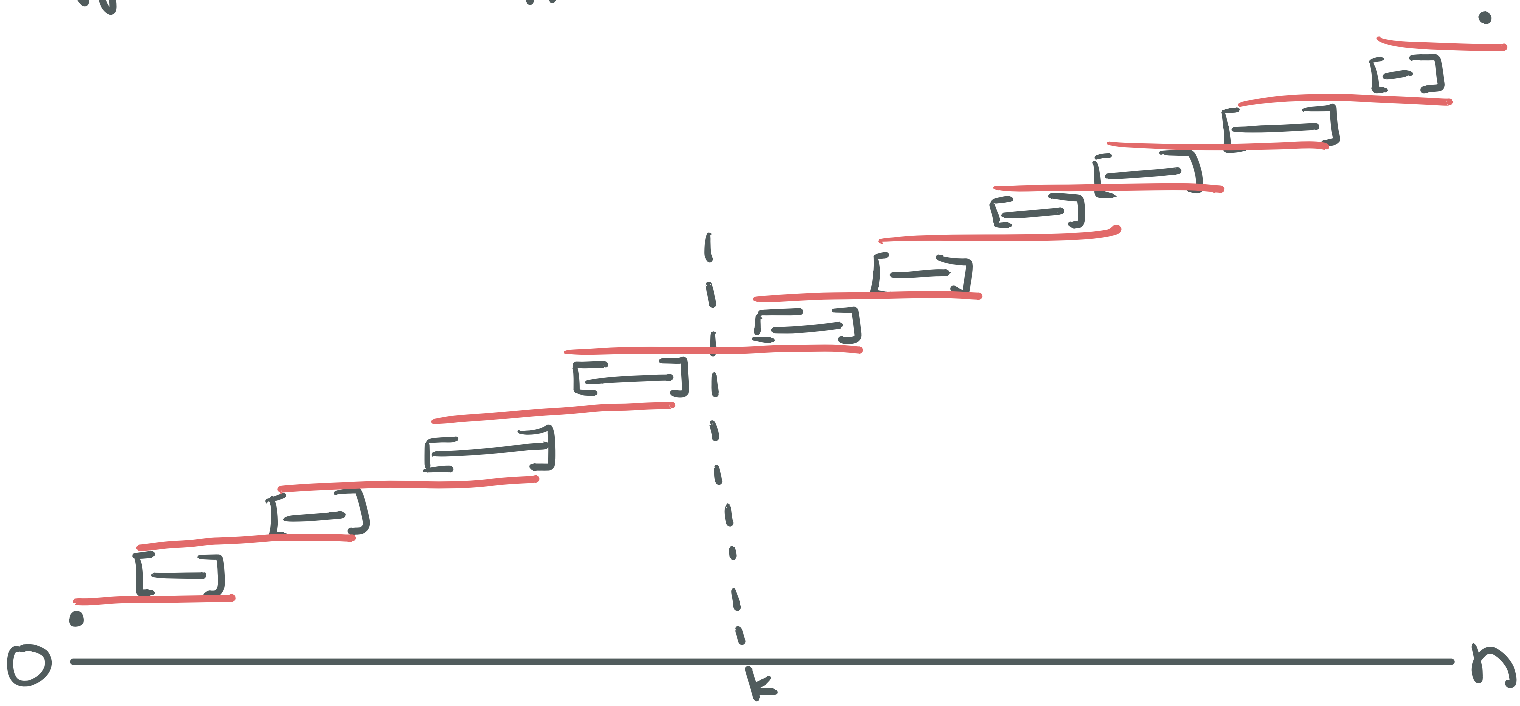
Suppose $k \in I(q_i)$ for some i

if $I(q_i) \subseteq [k-\varepsilon n, k+\varepsilon n]$ then done

else look at q_{i-1}

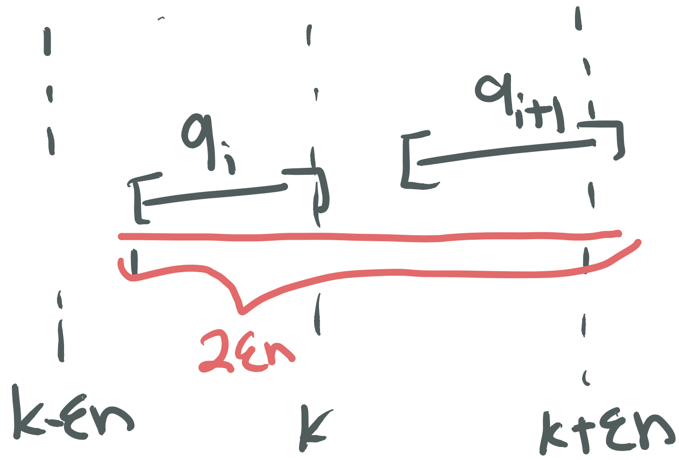


suppose $k \notin I(q_i)\ \forall i$



the "combined intervals" cover $[n]$

pick 1 covering k.

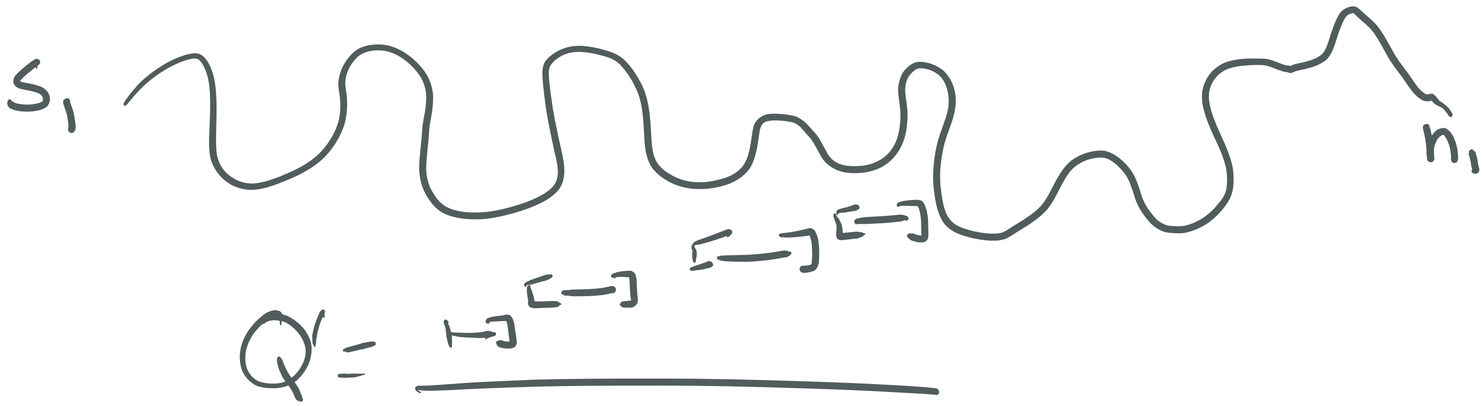


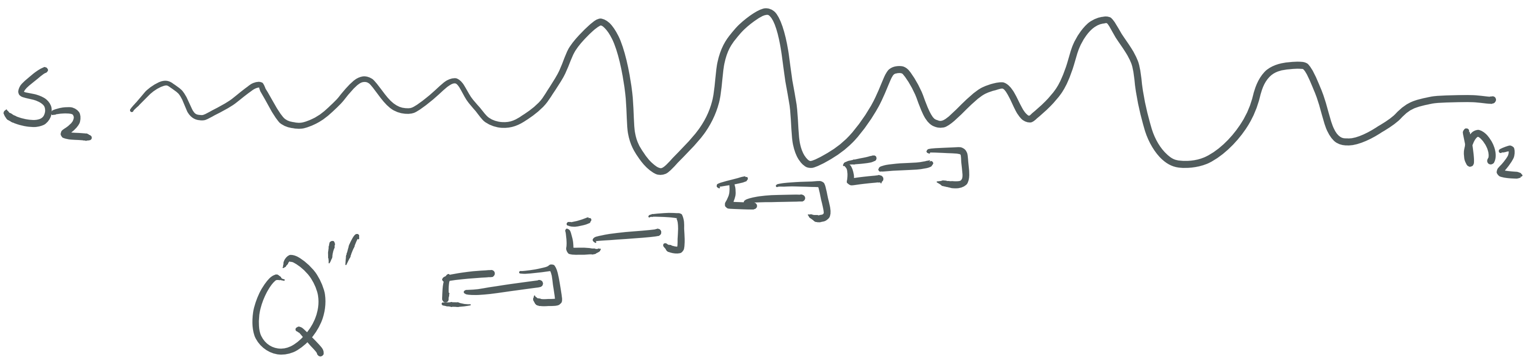
one of the intervals must lie inside

Key invariant: any two consecutive intervals have width $\leq 2\varepsilon n$.

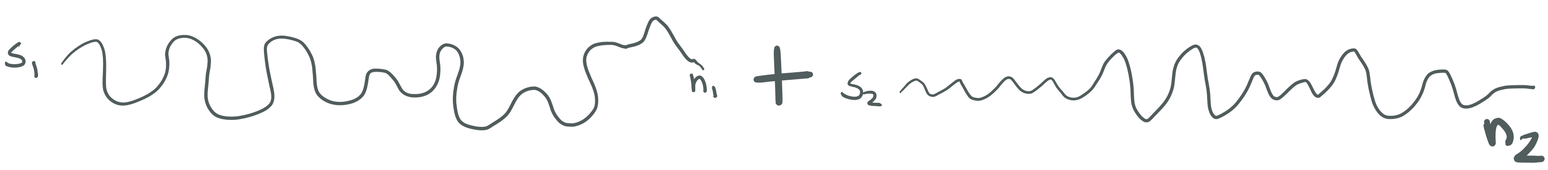
"ε-APX quantile summary"

<u>Merging</u> given two ε-APX quantile summaries over 2 streams, want ε-APX summary over <u>combined stream</u>





want to combine Q', Q'' to get summary of



$$\text{“}Q' + Q''\text{”} = \{q_1''', \ldots, q_c''', I'''(q_1'''), \ldots, I'''(q_c''')\}$$

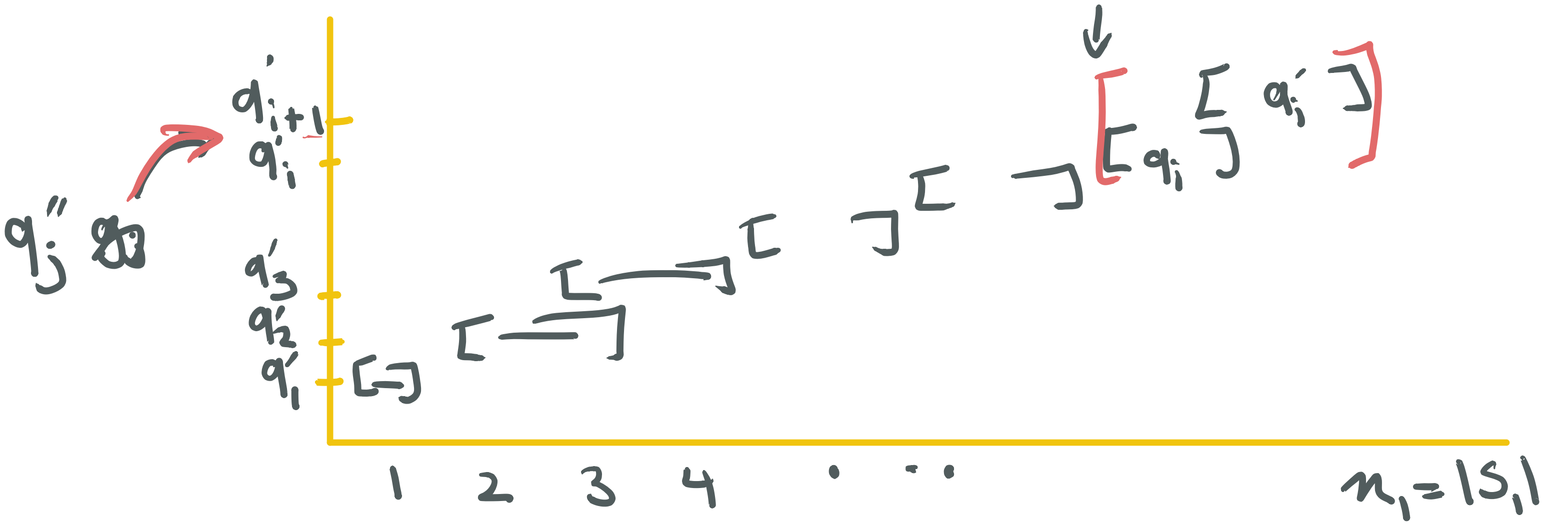
denote:

$$Q' = \{q'_1, \dots, q'_\ell, I'(q'_1), \dots, I'(q'_\ell)\}$$

$$Q'' = \{q''_1, \dots, q''_m, I''(q''_1), \dots, I''(q''_m)\}$$

let $q''_j \in Q''$. $I''(q_j)$ bounds rank q''_j w/r/t S_2

goal: bound rank q''_j w/r/t $S_1 + S_2$.



rank of q''_j in S_1 $\begin{cases} \geq \min I'(q'_i) \\ \leq \max I'(q'_{i+1}) \end{cases}$

so $\min I'(q'_i) + \min I''(q''_j)$

$$\leq \text{rank}(q''_j \text{ in } S_1 + S_2)$$

$$\leq \max I'(q'_{i+1}) + \max I''(q''_j)$$

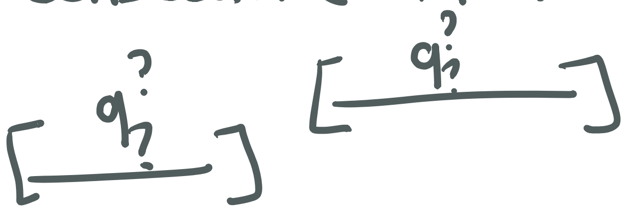
set $I'''(q''_j) = [\min I'(q'_i) + \min I''(q''_j),$

$\max I''(q'_{i+1}) + \max I''(q''_j)]$

$Q''' = \{q'_1, \dots, q'_\ell, q''_1, \dots, q''_m$, w/ intervals $I'''\}$

to show Q''' is ε-APX, need to show "$2\varepsilon n$ width" property.

Take two consecutive intervals in Q'''_3.

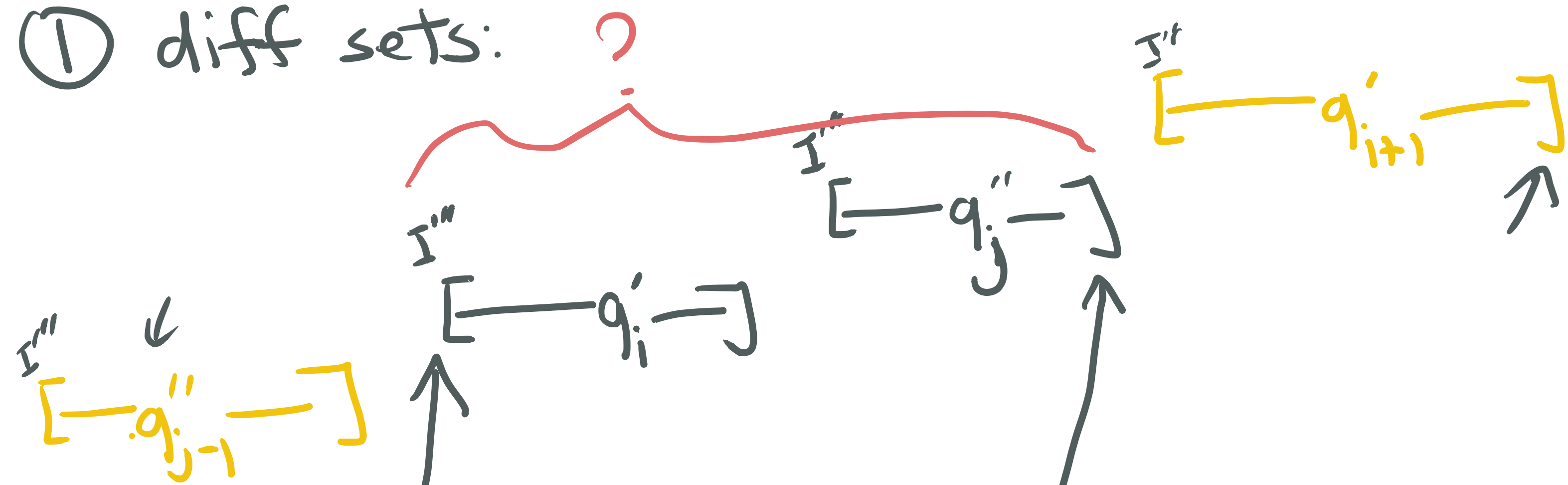


Two cases:

1. elements from diff sets

2. elements from same sets

① diff sets:



$\min I'(q'_i) + \min I''(q''_{j-1})$ $(\ast)$

$\max I'(q'_{i+1}) + \max I''(q''_{i+1})$ $(\ast\ast)$

$$(\ast) - (\ast\ast) = \begin{array}{l} \boxed{\max I'(q'_{i+1})} + \boxed{\max I''(q''_{j+1})} \\ - \boxed{\min I'(q'_i)} + \boxed{\min I''(q''_{j-1})} \\ \hline \le 2\varepsilon n_1 \quad + 2\varepsilon n_2 \end{array}$$

$$= 2\varepsilon(n_1 + n_2)$$

② same sets

I'' $[-\, q_i' \,-]$ I''' $[-\, q_{i+1}' \,-]$ $[-\, q_{j+1}'' \,-]$

$[-\, q_j'' \,-]$

$\min I'(q_i') + \min I''(q_j'')$ $\max I'(q_{i+1}') + \max I''(q_{j+1}'')$

$$\begin{array}{r} \max I'(q_{i+1}') + \max I''(q_{j+1}'') \\ -\ \min I'(q_i') + \min I''(q_j'') \\ \hline 2\varepsilon n_1 + 2\varepsilon n_2 = 2\varepsilon(n_1+n_2) \end{array}$$

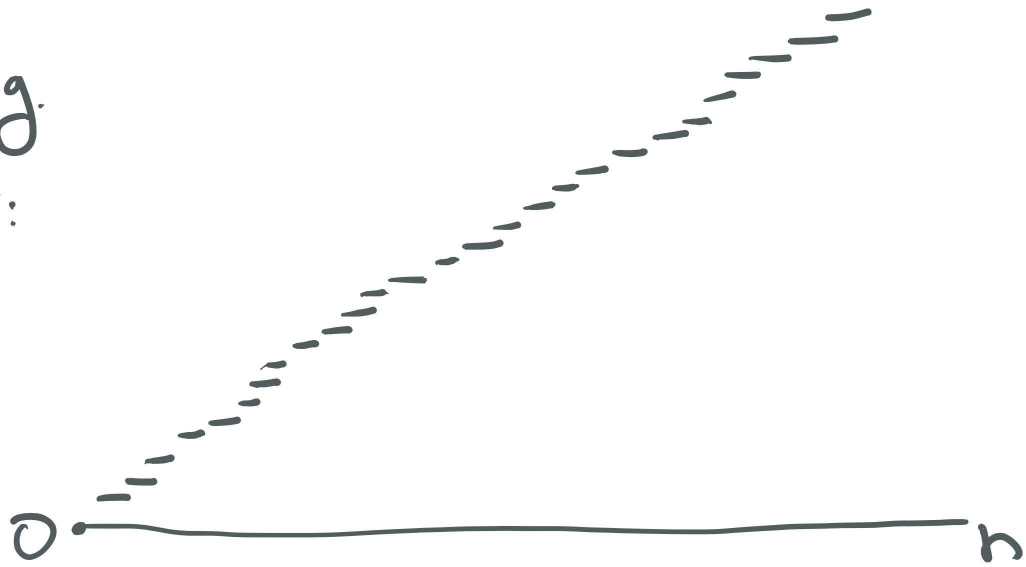
$\blacksquare$

This shows that merging 2 ε-APX QS's gives ε-APX QS of combined streams

size?

Pruning

Input:

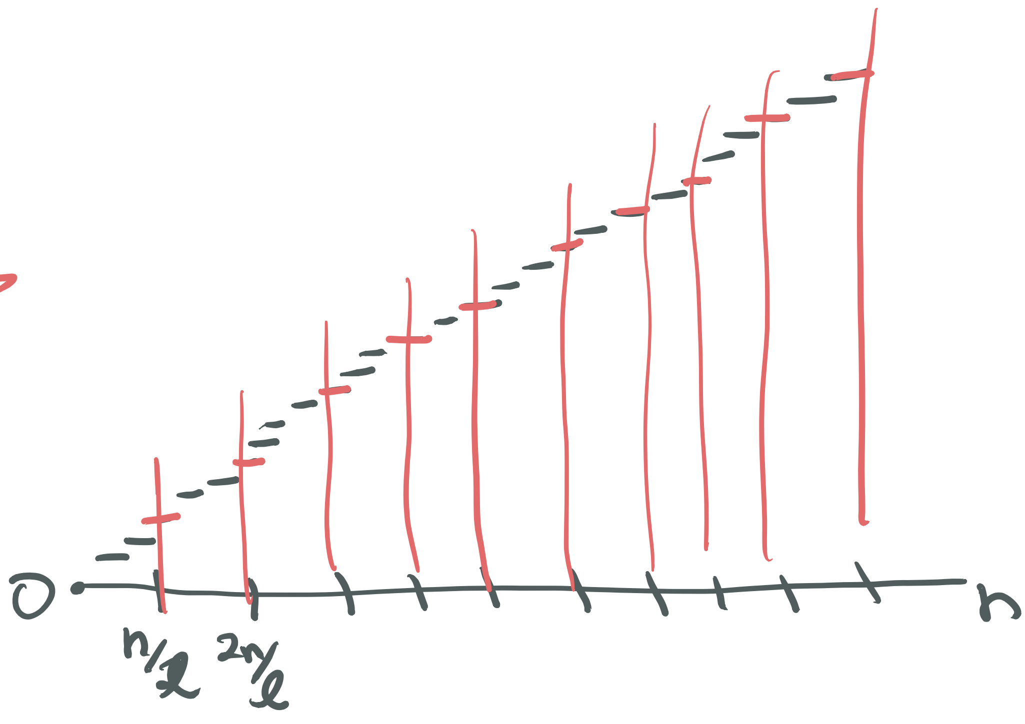


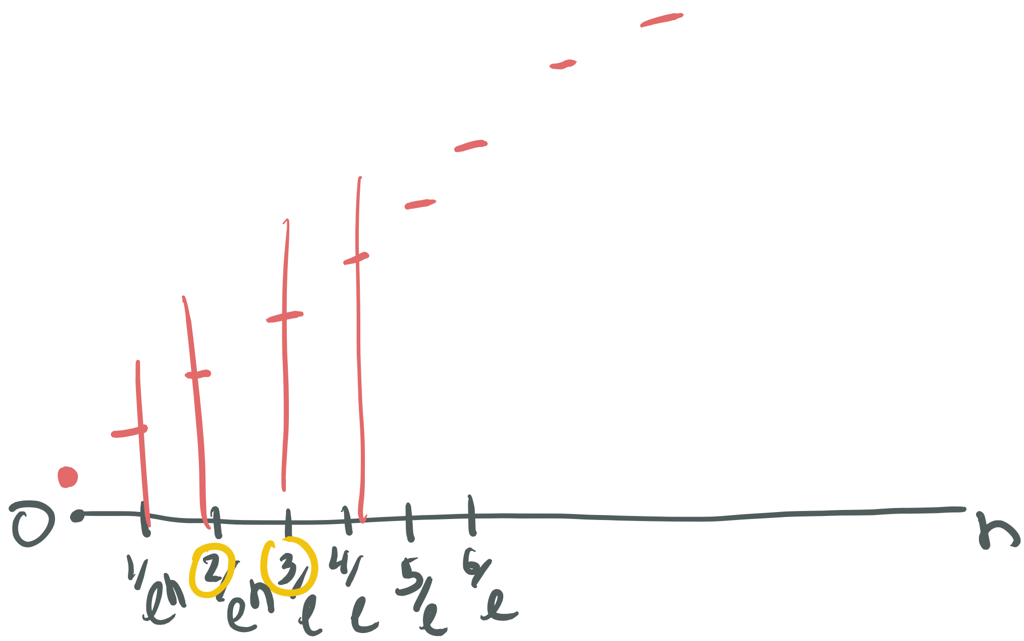
ε-approximate quantile summary w/ too many points

Goal: sparser summary that's still very good

~~g~~ keep

ℓ queries



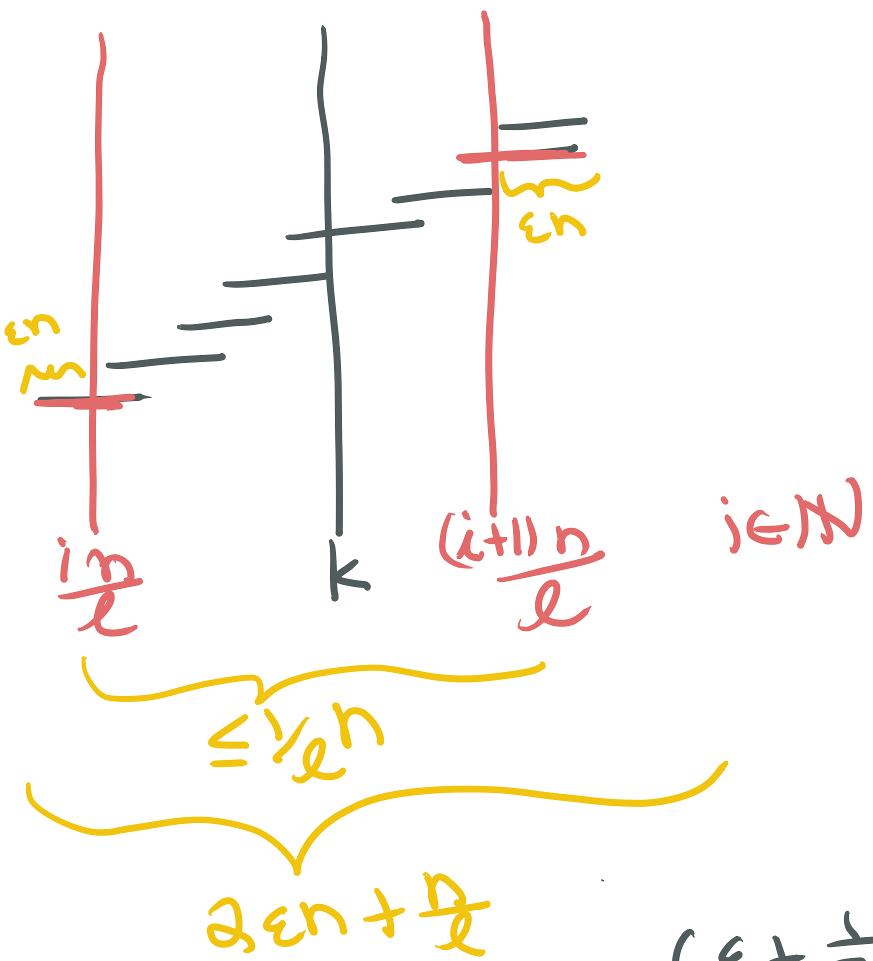


Claim: resulting quantile is $(\varepsilon + \frac{1}{2\ell})$-APX

Proof

suppose we query a rank k

look at original summary



$(\varepsilon + \frac{1}{2\ell})$-Approx

Recap: we can

combine ε-APX quantile summaries
to get ε-APX quantile summary of whole thing

sparsify ε-APX quantile summary to
$(\varepsilon+\frac{1}{2k})$-APX quantile summary w/ k points

Remains to address:

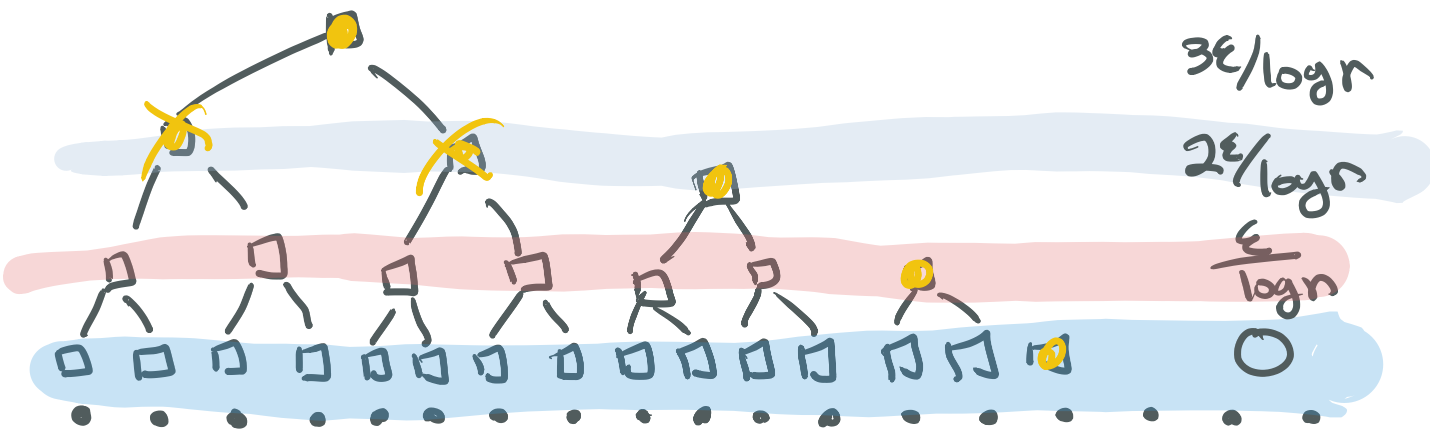
how to make one at all??

what if $n=1$?

Take the point

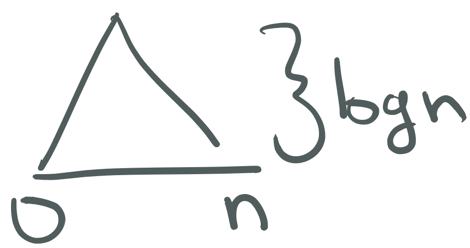
I claim that's all we need!

APX



take $k = \cancel{\frac{\varepsilon}{\log(n)}}\ \frac{\log n}{2\varepsilon}$

at the root,



ε-approximate quantiles

Space?

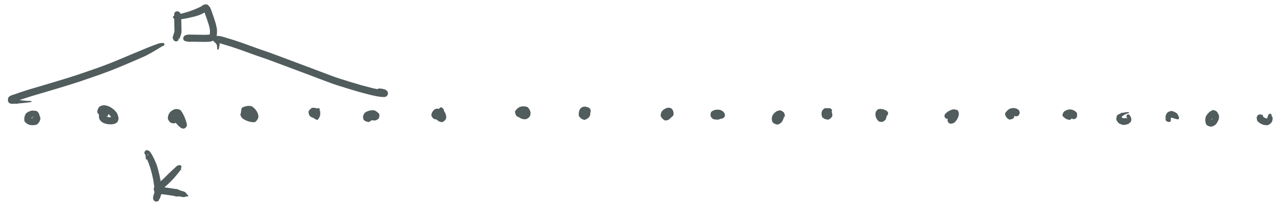
only keep "root summaries"

Theorem

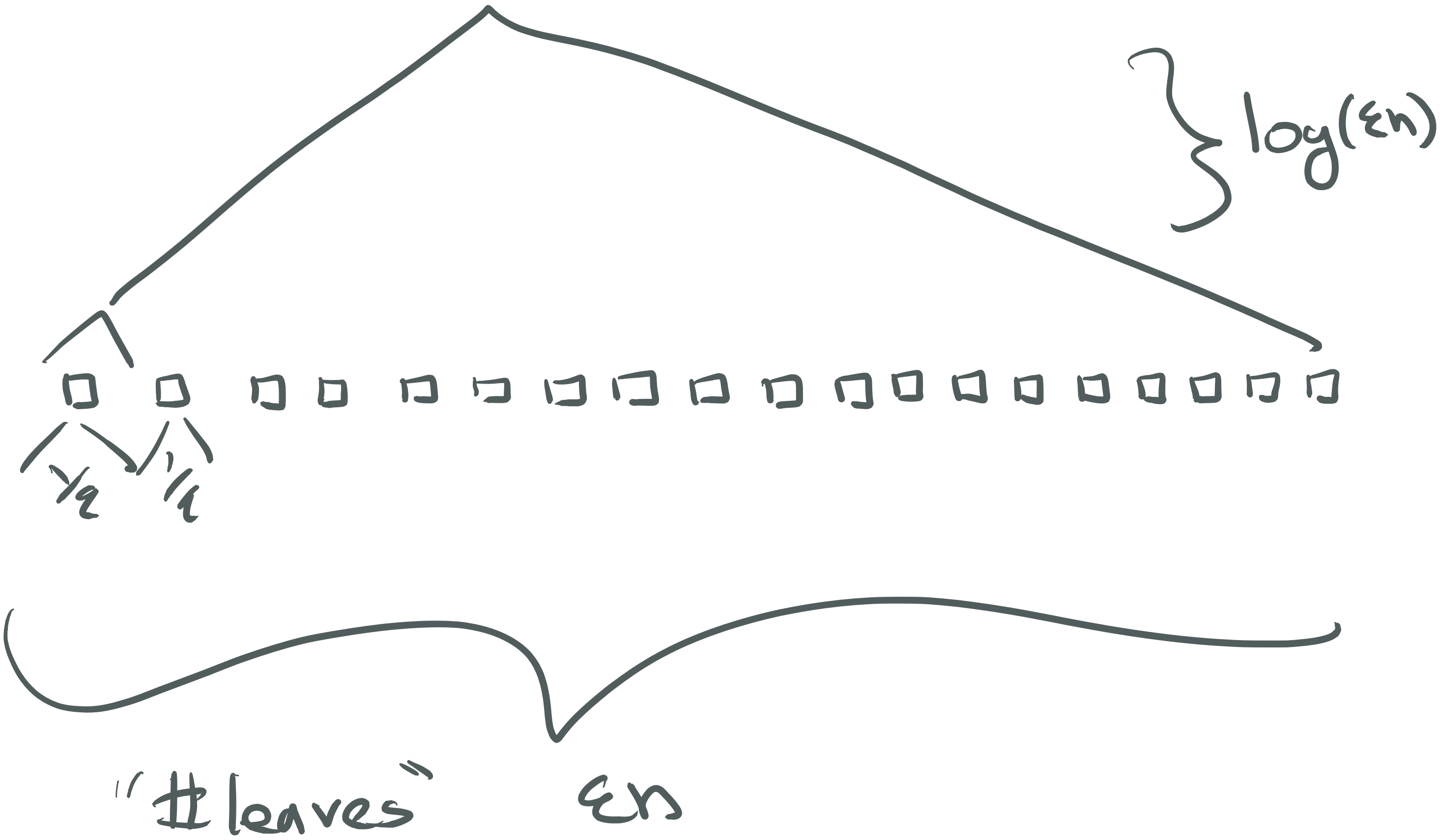
- 1-pass
- $O(\log^2(n)/\varepsilon)$ space
- deterministic
- ε-APX quantile over stream

idea: mergability + dyadic intervals trick

slightly better:



have first level contain $1/\varepsilon$ points



Theorem ††

- 1-pass
- $O(\log^2(\varepsilon n)/\varepsilon)$ space
- deterministic
- ε-APX quantile over stream

Even better?

Khanna-Greenwald [2001]:

$\frac{1}{\varepsilon}\log(\varepsilon n)$ space

- more sophisticated quantile summary, merging
- interval trick

Finding the median (and other ranks) in p passes

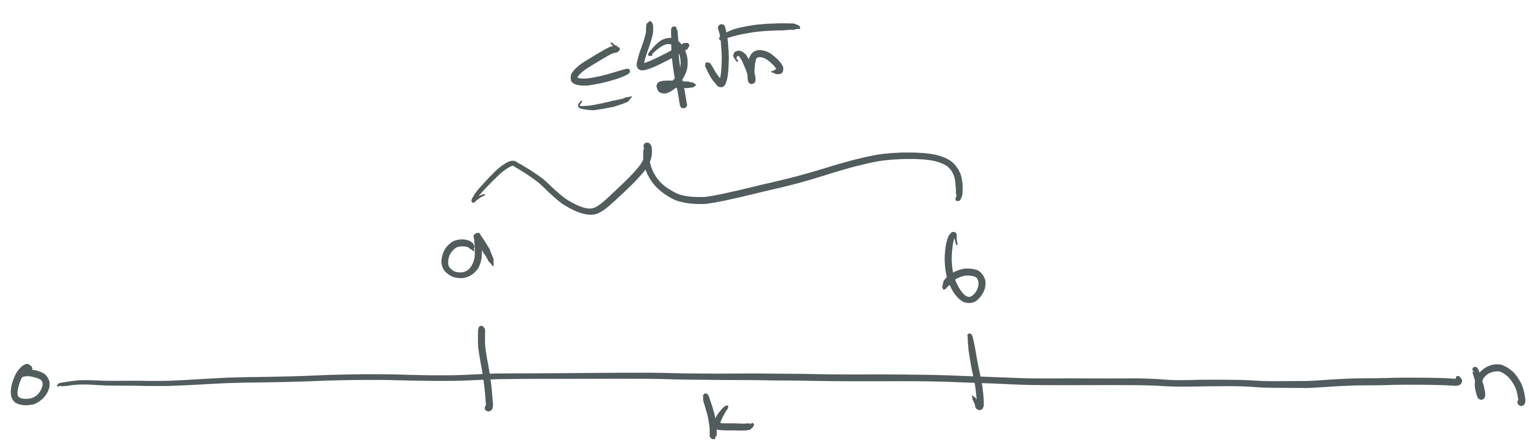
Fix $p=2$ for simplicity.

goal: $O(\sqrt{n}\ \text{polylog}(n))$ space

suppose we are querying rank k.

1st pass: build ε-APX quantile summary for $\varepsilon = 1/\sqrt{n}$ ($\sqrt{n}\log(n)$ space w/ GK)

query $k-\sqrt{n}$, $k+\sqrt{n} \Rightarrow a, b$

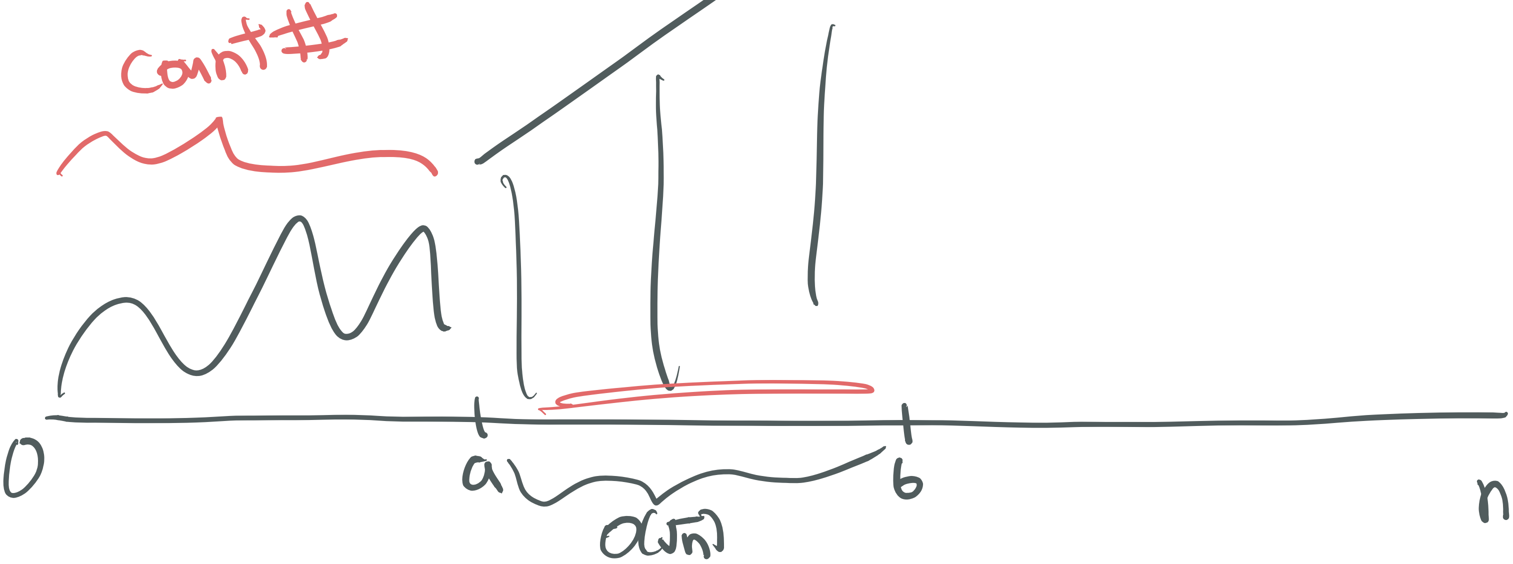


$$\underline{k-2\sqrt{n} \le \text{rank}}(a) \underline{\le k} \le \text{rank}(b) \le k+2\sqrt{n}$$

$a = \text{query}(k-\sqrt{n})$

2nd pass:

•



k over all

Take the $(k - \#\{<a\})$th in the sorted set

for general p:

make $\frac{1}{n^{1/p}}$-APX quantile summaries and filter.

After p passes, down to $n^{1/p}$ elements sort and select.