Range Min Query (RMQ)

Static: given array $a_0, a_1, \ldots, a_{n-1}$

findMin$[l, r) = \min\{a_l, a_{l+1}, \ldots, a_{r-1}\}$.

Dynamic: also support

change(pos, value) // $a_{pos} :=$ value;

To speed up queries, we can do preprocessing

$\langle O(f(N)), O(g(N)) \rangle$ means $O(f(N))$ time for preprocessing and $O(g(N))$ per query

$\langle O(1), O(N) \rangle$ – Naive Loop

$\langle O(N^2), O(1) \rangle$ – precalc all ranges

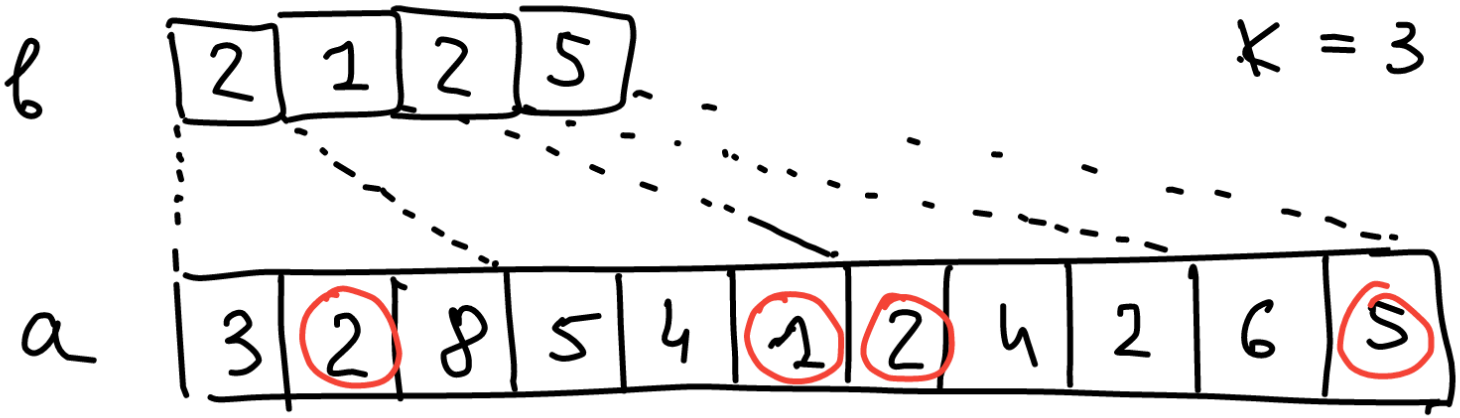
$\binom{n}{2} = \frac{n(n-1)}{2} \in \Theta(N^2)$ pairs $0 \le l < r < N$

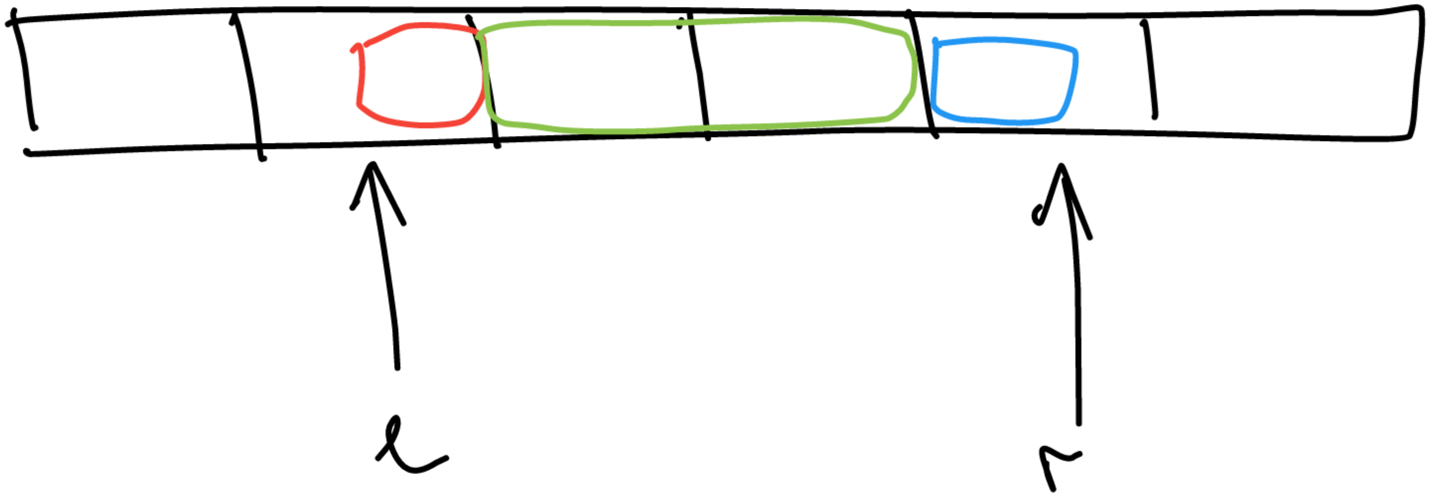
Square root decomposition:

$k = \Theta(\sqrt{N})$

$b_i = \min\{a_{ki}, a_{ki+1}, \dots, a_{ki+k-1}\}$

$K = 3$





Query can be split in 3 parts:

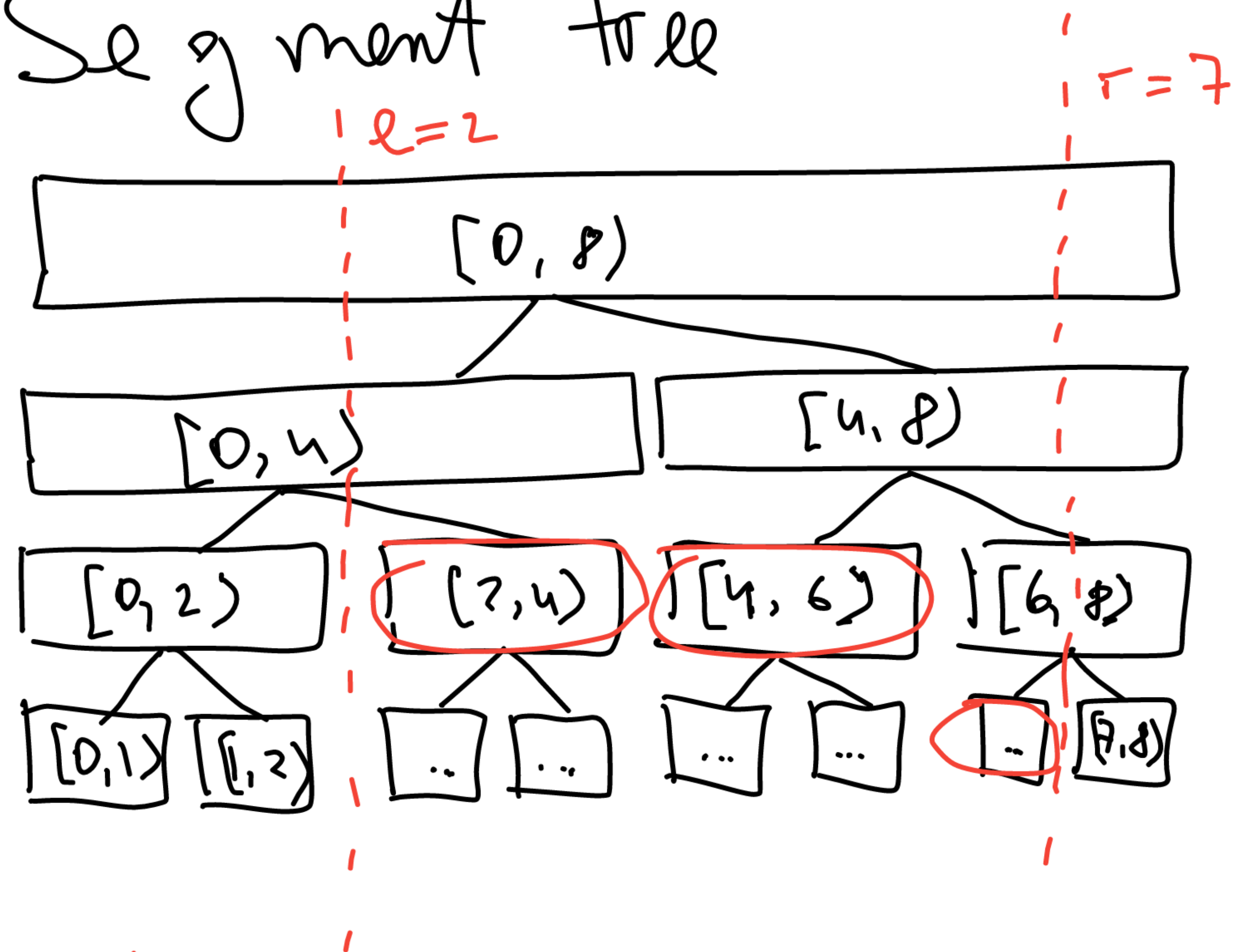
$\leq K + \frac{N}{K} + K \in O(\sqrt{N})$ elements

$\langle O(N), O(\sqrt{N}) \rangle$ data structure.

Bonus: can support change() in $O(\sqrt{N})$

Use this idea $\log_2 N$ times:

Segment tree



$[2,7) = [2,4) + [4,6) + [6,7)$

Segment tree

Lemma 1: any $[l, r)$ can be split in no more than $2 \log_2 N$ segments.

Lemma 2: there are $O(N)$ nodes

$\langle O(N), O(\log N) \rangle$ data structure for RMQ

Bonus: supports changes in $O(\log N)$

(update leaf value & all parents)

How to get $O(1)$ query?

Only static setting.

$\circ : X \times X \to X$ is idempotent, if

$$x \circ x = x$$

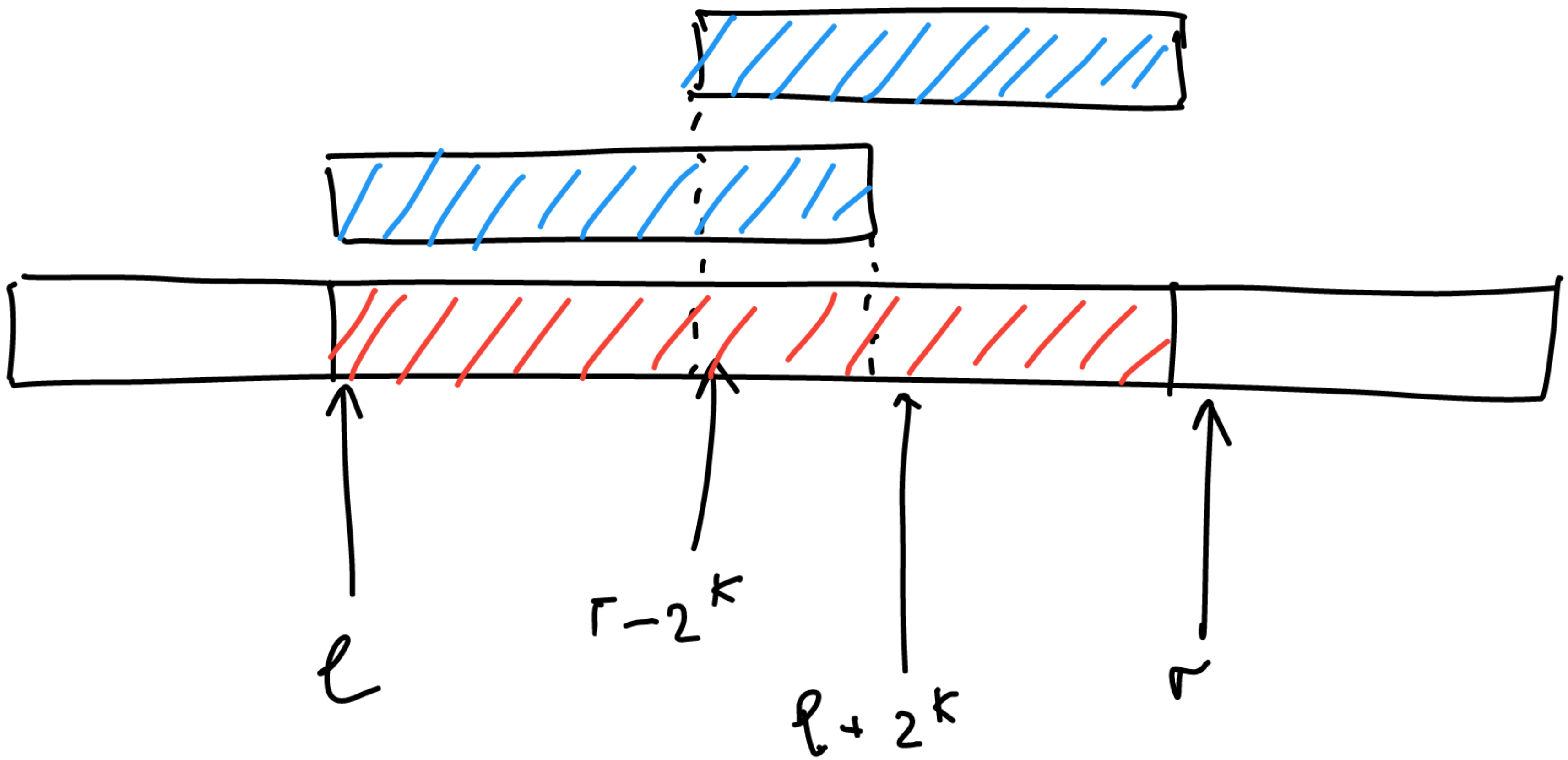
Examples: $\min(x, x) = x$ $\max(x, x) = x$

Counter examples: $x + x \neq 2x$ in general

Sparse Table idea

for any $[l, r)$ $\exists k$:

$$[l, r) = [l, l+2^k) \cup [r-2^k, r)$$



$$b_{k,i} = \min\{a_i, a_{i+1}, \dots, a_{i+2^k-1}\}$$

$$b_{0,i} = a_i$$

$$b_{k,i} = \min\{b_{k-1,i}, b_{k-1,i+2^{k-1}}\}$$

$$\text{findMin}[l, r) = \min\{b_{k,l}, b_{k,r-2^k}\}$$

How much memory for b?

$$O(\log N) \times O(N) = O(N \log N)$$

How to find k?

$S = r - l$ elements in $[l, r)$

$2^k \le S$!

take maximal k that $2^k \le S$

$\underline{0\,0\,1\,0\,0\,0}_2$	$S = 8$	$2^k = 8$	$k = 3$
$0\,0\,1\,0\,1\,1_2$	$S = 11$	$2^k = 8$	$k = 3$
$0\,0\,1\,1\,1\,1_2$	$S = 15$	$2^k = 8$	$k = 3$
$0\,1\,0\,0\,0\,0_2$	$S = 16$	$2^k = 16$	$k = 4$

$$k = \text{getHighestBit}(r - l)$$

$$k = \lfloor \log_2 (r - l) \rfloor$$

How to find k quickly?

Brute force: $O(\log N)$

Binary search: $O(\log \log N)$

$$\left.\begin{array}{l}
k = 0; \\
\text{if } (s \geq 2^{16}) \ \{ k \mathrel{+}= 16; \ s = s / 2^{16}; \} \\
\text{if } (s \geq 2^{8}) \ \{ k \mathrel{+}= 8; \ s = s / 2^{8}; \} \\
\text{if } (s \geq 2^{1}) \ \{ k \mathrel{+}= 1; \ s = s / 2; \}
\end{array}\right\} \log 32$$

Precalculate $\log_2 S$

$$\log_2 1 = 0$$

$$\log_2 S = 1 + \log_2 \frac{S}{2}$$

$$\left\lfloor \log_2 S \right\rfloor = 1 + \left\lfloor \log_2 \left\lfloor \frac{S}{2} \right\rfloor \right\rfloor$$

$$lb(S) = 1 + lb\left(\left\lfloor \frac{S}{2} \right\rfloor\right)$$

$\langle O(N), O(1) \rangle$ to find k

All together makes

$\langle O(N \lg N), O(1) \rangle$

Static RMQ

using Sparse Table

Memory $\in O(N \lg N)$

The ultimate goal: $\langle O(N), O(1) \rangle$

RMQ (to be continued ...)