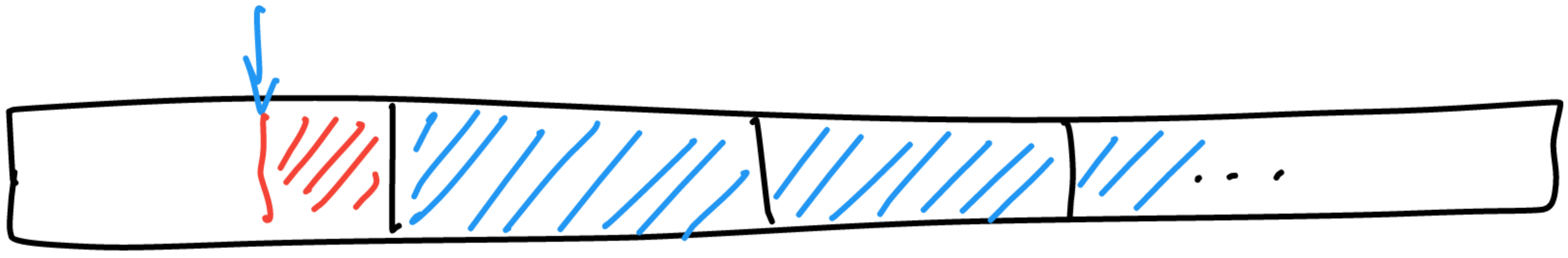
1



$B = \log_2 N$

SparseTable(block)

Global Sparse Table keeps block minimums

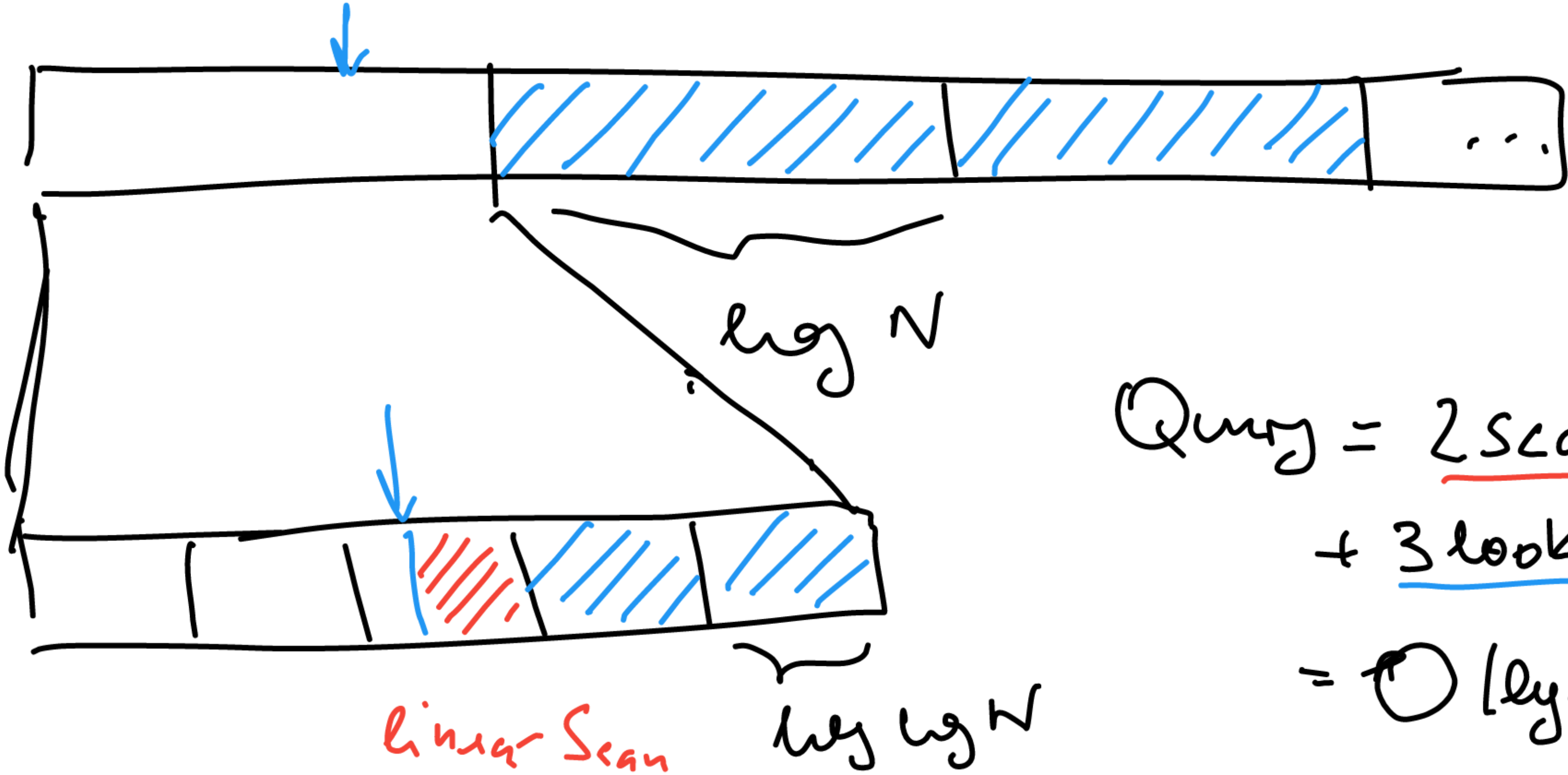
$$O\left(\frac{N}{\log N} \cdot \log\left(\frac{N}{\log N}\right)\right) = O(N)$$

$$+ \text{ ST per each block: } \frac{N}{\log N} \cdot O(\log N \log \log N)$$

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

Query = 1 global + 2 local = $O(1)$

Optimization: Store 1 ST with depth $O(\log \log N$

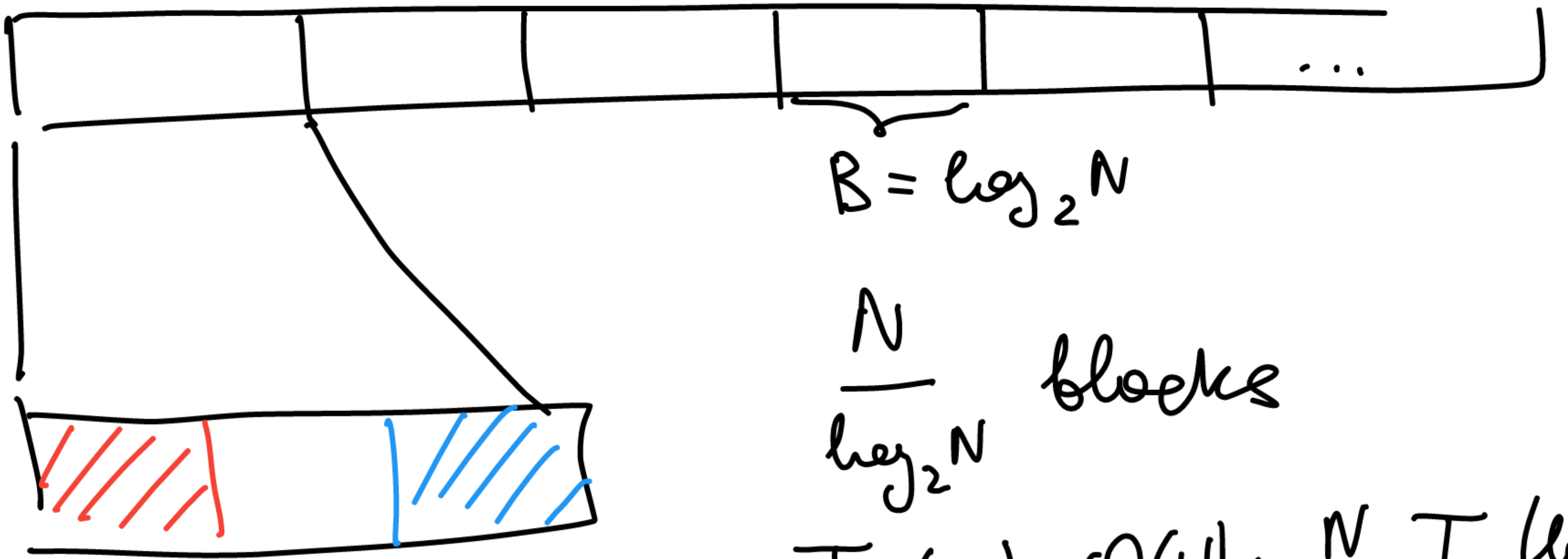


Query = 2 scans + 3 lookups = $O(\log\log N)$

$$O\left(\frac{N}{\log N}\cdot \log\left(\frac{N}{\log N}\right)\right) = O(N)$$

$$\frac{N}{\log N}\cdot O\left(\frac{\log N}{\log\log N}\cdot \log\left(\frac{\log N}{\log\log N}\right)\right) = O(N)$$

optimization: store 2nd level in 1 table



$B = \log_2 N$

$\frac{N}{\log_2 N}$ blocks

$T_B(N) = O(N) + \frac{N}{\log N} T_B(\log n)$

Build recursively!

+ suffix min

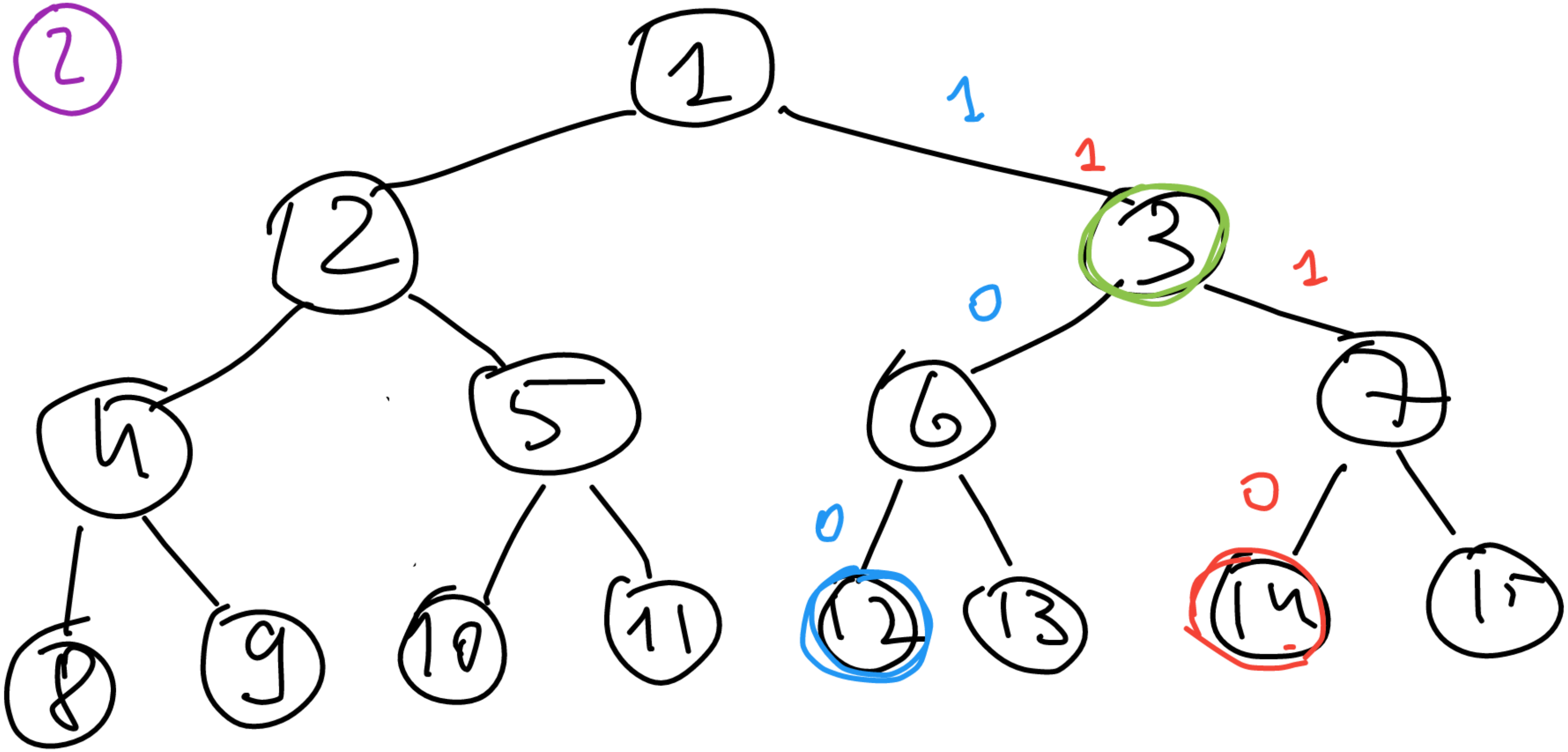
+ prefix min

$T_Q(N) = O(1) + T_Q(\log N)$

without suffix / prefix:

$T_Q'(N) = O(1) + 2 T_Q'(\log N) = O(2^{\log^* N})$

2



$12 = \boxed{11}00_2$

$14 = \boxed{11}10_2$

$3 = 11_2$

$u \operatorname{xor} v = \underbrace{00\ldots0}_{\text{LCA bits}}1\ldots.$

position of highest bit

③ $b = \frac{1}{2}\log_2 N$. Store answer for all masks of length $1, 2, \dots, b$. Total $2^b + 2^{b-1} + \dots + 2 + 1 = 2^{b+1} - 1 =$
$= O(\sqrt{N})$ masks.

mask $= 1 \quad \boxed{1 \quad 0 \quad 0 \quad 0 \quad 1}$

$a = 0, 1, 2, 1, 0, -1, 0$

mask$' = \boxed{1 \quad 0 \quad 0 \quad 0 \quad 1}$

$a' = 0, 1, 0, -1, -2, -1$

$$F(mask) = \min(0, F(mask') \pm 1)$$

+ or - ?

$$mask' = mask \,\&\, (2^{k-1} - 1)$$

We need to extract leftmost bit:

$$mask \geq 2^{k-1} = \underbrace{100\ldots0}_{k \text{ bits}}{}_2$$

Or just reverse mask $\longrightarrow$

$mask \bmod 2$: rightmost bit

$$mask' = mask / 2$$

4 Calculate jump pointers after adding a vertex u:

$$jump[u, 0] = par[u]$$

$$jump[u, k] = jump[jump[u, k-1], k-1]$$

$O(\log N)$ per vertex

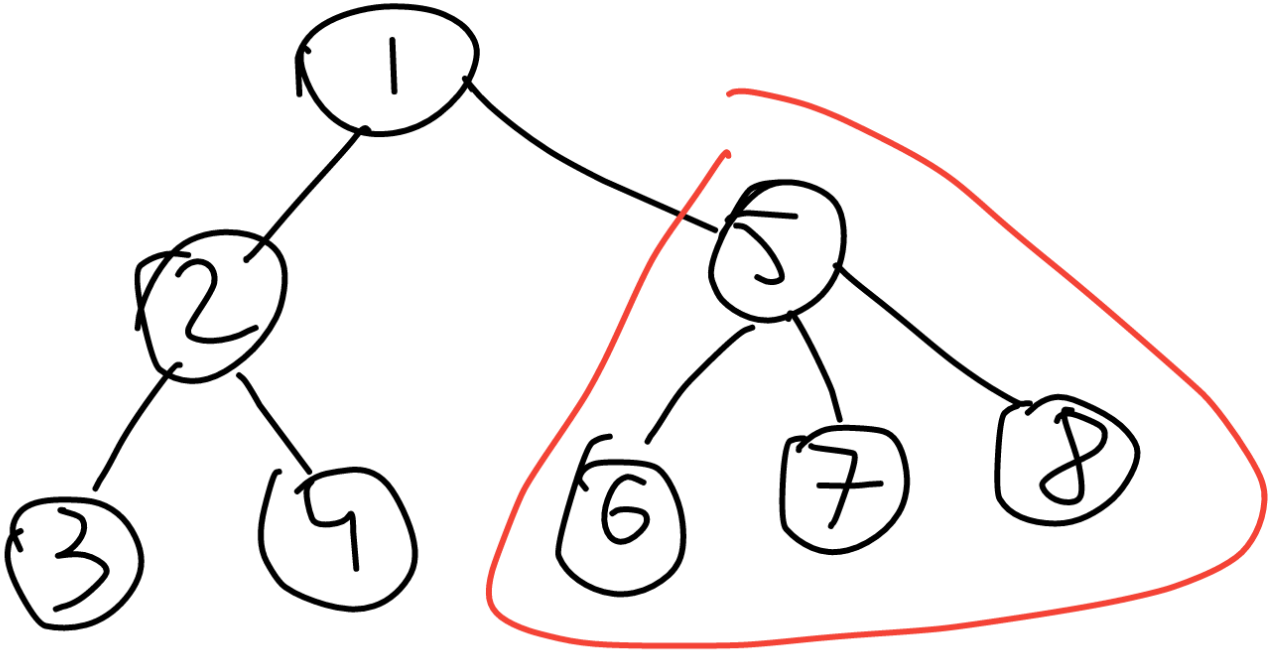
5 Jump pointers. Store

$\min(a[u], a[par[u]], \ldots, a[par^{k-1}[u]])$

with jump pointer k

$$par^{k-1}[u] = \underbrace{par[par[\ldots pa[u] \ldots]]}_{k-1 \text{ times}}$$

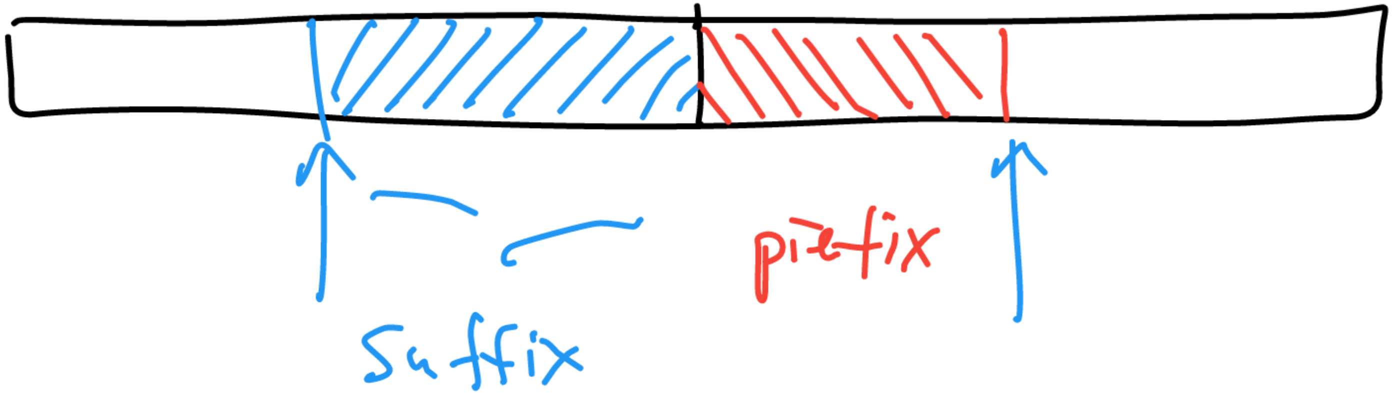
6 Euler Tour. Store vertex once

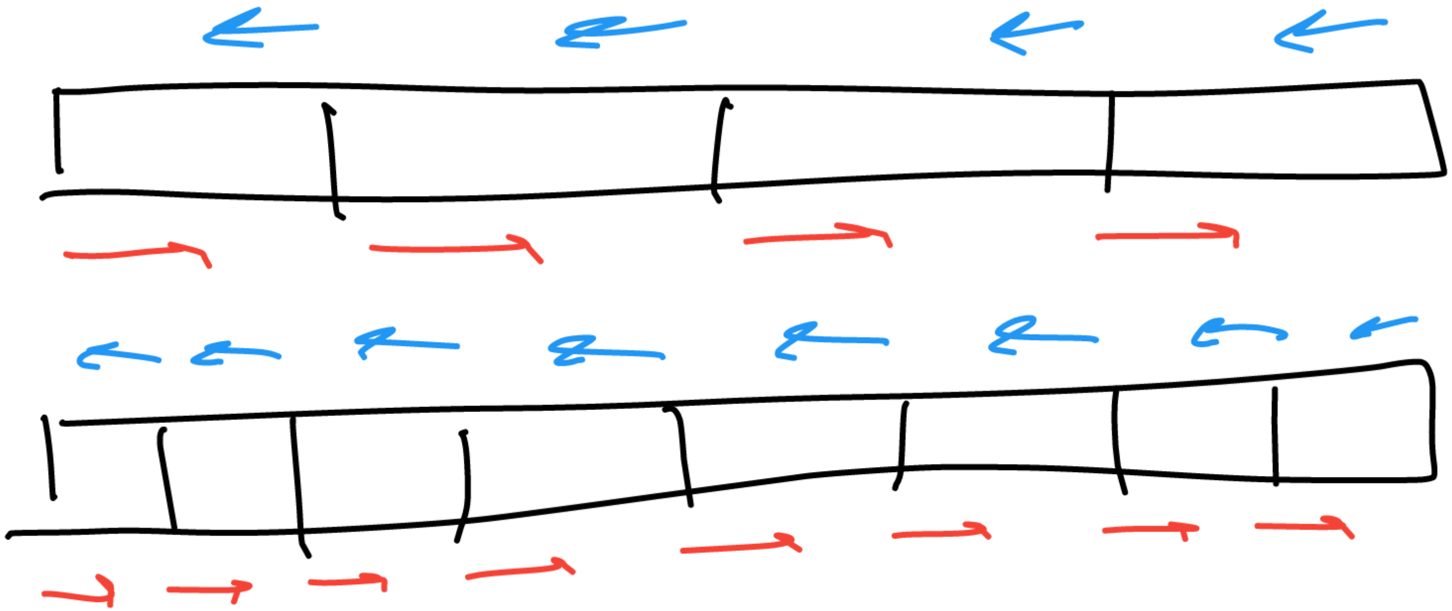


1 2 3 4 5 6 7 8

Then Segment Tree or BST to store set of colored vert!

⑦ $R - L \geq N/2$:





Level - highest bit of R-L

level = 2

$$\overbrace{0\,1}\,0\,1\,1_2$$

$$\underbrace{1\,0}$$

K = 3