Sparse Tables

→ One of the most easiest DS to code.

→ Solves a variety of problems in a super optimized manner.

IDEMPOTENT func^n

$$f(f(x)) = f(x)$$

Ex → $abs(x)$ ⟶ $abs(abs(x)) = abs(x)$

Ex → $f(x) = x * 0$ ⟶ $f(f(x)) = f(x)$

Then, it is a func^n that has no additional effect if it is called more than once on same parameters.

$\min(\min(4, 3), 3) \rightarrow \underline{\underline{3}}$

$\gcd(\gcd(x, y), y) \rightarrow \gcd(x, y)$

<u>But</u>

$\operatorname{sum}(\operatorname{sum}(x, y), y) \;!= \operatorname{sum}(x, y)$

$\operatorname{sum}(x, y) \rightarrow$ is not an idempotent func^n.

$$\max\left(\min(l, r'),\ \min(l', r)\right)$$

$$= \min(l, r)$$

Q

int array

Range queries → L, R

↓

RMQ → Range min query.

In what time complexity you can answer a query of min(l, r)

$\sqrt{n}$ → sqrt decom
log n → segment tree
log n → BIT
1 → sparse table

with a catch

1. Sparse table can answer queries in O(1) time for idempotent func^n.

2. For Non-idempotent func^n it take O(log n)

$\boxed{\text{Sum}} \rightarrow O(\log n)$

$\boxed{\text{min, max, gcd}} \longrightarrow O(1)$

fact → Every no. can be represented as sum of power of 2.

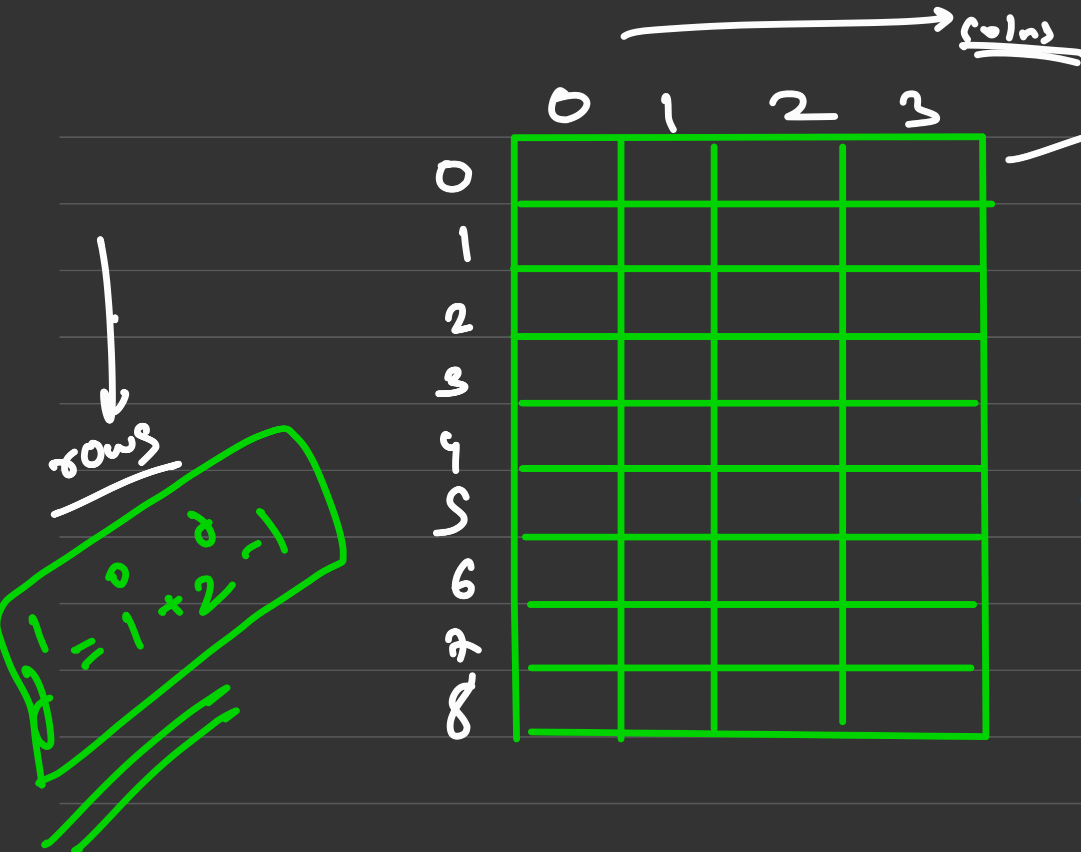
1101101 → sum(power of 2's)

$13 \rightarrow 2^3 + 2^2 + 2^0$

→ sparse table pre-computes ans of a lot of ranges. by creating a sparse matrix :-

dimension → $n \times m$

$$\left\{ \begin{array}{l} n \rightarrow \text{max length of query} \\ m \geq \lceil \log_2 n \rceil + 1 \end{array} \right\}$$



tb[i][j]

starting at the i^{th} index and querying for j' length

7	2	3	0	5	10	3	12	18
0	1	2	3	4	5	6	7	8

	0	1	2	3
0	7	2	0	0
1	2	2	0	0
2	3	0	0	x
3	0	0	0	x
4	5	5	3	x
5	10	3	3	x
6	3	3	x	x
7	12	12	x	x
8	18	x	x	x

$0,0 \rightarrow [0,0]$

$1,0 \rightarrow [1,1]$

$2,0 \rightarrow [2,2]$

$0,1 \rightarrow [0,1]$

$1,1 \rightarrow [1,2]$

$0,2 \rightarrow [0,3]$

$4,2 \rightarrow [4,7]$

$(i,j) \rightarrow [i, i+2^j-1]$

$0,3 \rightarrow [0,7]$

$O(n \log n)$ → to build

7	2	3	0	5	10	3	12	18
0	1	2	3	4	5	6	7	8

$n \rightarrow n^2$ ranges possible.

→ In first column, we store ans of all queries possible of layer 1

→ In second column we store ans of all queries of layer 2.

→ We are storing ans of all the ranges viz a power of 2 in layer in the table.

	0	1	2	3
0	7	2	0	
1	2	2	0	0
2	3	0	0	X
3	0	0	0	X
4	5	5	3	X
5	10	3	3	X
6	3	3	X	X
7	12	12	X	X
8	18	X	X	X

RMQ

(1) 2, 7

(2) 1, 6

(3) 3, 8

(4) 2, 3

$L = 8$

(1) $L = 7 - 2 + 1 \rightarrow 6$

$l = 4$

$2\ [4, 4 + 2 - 1]$

6

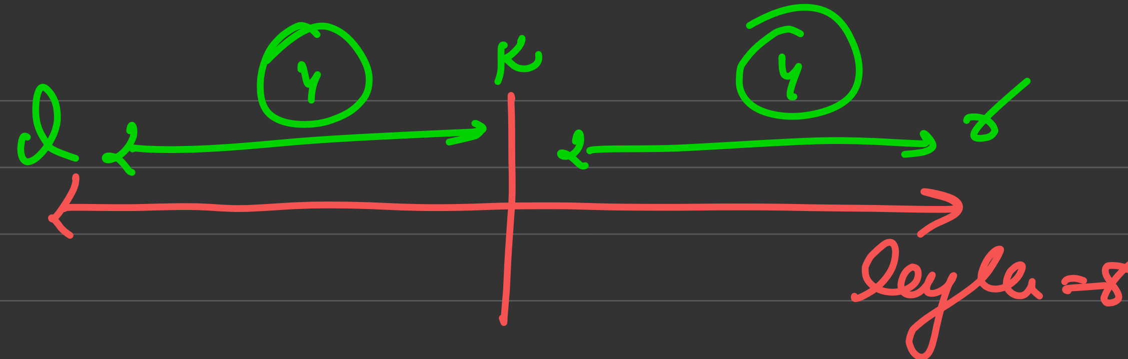
$j = 1$

$j = 0$

$[0, 0 + 3]$

7	2	3	0	5	10	3	12	18
0	1	2	3	4	5	6	7	8

$l \mathrel{+}= 2^1 \rightarrow 4 + 2 = 6$



4 is a power of 2

$$\min(l, r) = \underbrace{\min(\min(l, k), \min(k+1, r))}_{O(1)}$$

2^j

i ⟶ k ⟶ 6

layer → 6

binlog → 2

(kecal) → 3

curr → 3

2 ⟷ 7

7 → 4 + 2 + 1 ⟶

given any length n, if the func^n is idempotent, then

(i) Calculate the power of α, viz just smaller than n.

7 2 3 0 5 10 3 12 18

0 1 2 3 4 5 6 7 8

2^k 2^k

$(0, 6) \to 2$ 6 − 4 + 1

2 + 1

→ 3

$l, r \to \text{len} \to r - l + 1$

$K \to \log 7 \to 2$ $k \to$ power of 2 just smaller than length.

$$f(l, r) \to f\left(f'(l, k), f'(r - 2^k + 1, k)\right)$$

$$f(0, 6) \to f\left(f'(0, 2), f'(3, 2)\right)$$

$n \to$

$0 \to X$

$1 \to 0$

$2 \to 1$

$3 \to 1$

$4 \to 2$

$5 \to 2$

$6 \to 2$

$7 \to 2$

$8 \to 3$

prefix

$f(i) = f(i/2) + 1$

$O(n)$ $O(n\log n)$

$O(q)$

$\log(i)$

0

$i \longrightarrow r \to 16$

17

Length → 17

K → 4

(i, j)

↓

$[i, i + 2^j - 1]$

$[0, 4]$

↓

$[0, 0 + 15]$

$[1, 4]$

↓

$[1, 16]$

$16 - 16 + 1$

1

$$f(l, r) = f\left(f'(l, k),\ f'(r - 2^k + 1, k)\right)$$

$$f[f(0, 4),\ f(1, 4)]$$

3, 8 → lu → 6 → K → 2

$f'(3, 2)$ $\quad f'(8 - 2^2 + 1, 2)$

$[3, 3 + 2^2 - 1]$ $\quad (5, 2)$

$[5, 5 + 2^2 - 1]$

2, 3

$[3, 6]$ $\quad [5, 8]$

$2, 3 \rightarrow \quad lb \rightarrow 2 \quad \rightarrow k \rightarrow 1$

$[2, 1] \quad [3 - 2^1 + 1, 1]$

$(2, \cancel{1}) \quad (2, 1)$

$(2, 2 + 2^1 - 1) \quad (2, 2 + 2^1 - 1)$

$\underline{(2, 3)} \checkmark \quad \underline{(2, 3)} \checkmark$

0, 7 → dy → 8

k = 3

$f'(0,3)$

$f(7-2^3+1, 3)$

1su

$f'(0,3)$

$f'(0,3)$

min(

for non-idempotent func^n RSQ

7 → 4 + 2 + 1

$K = \log 7 \rightarrow K = 3$

$j = 3$

$(l, j) \rightarrow l += 2^j$

0 1 2 3 4 5 6

4 2 1