input ranges	query ranges	$k = 3$
① 1 – 7	5 – 7	accept if V in k ranges
② 2 – 6		
③ 3 – 10		
④ 2 – 4		
⑤ 6 – 8		

query (5 – 7)

5 = 1, 2, 3, 5 = 4 ranges (✓)

6 = 1, 2, 3, 5 = 4 ✓

7 = 1, 3, 5 = 3 ✓

for $k = 3$ 5 < 6, 7 are $ok = 3$

for $k = 4$ 5, 6 are $ok = 2$

Constraints as hints

200k inputs, 200k queries $(n = 200k)/nlogn$

either $\simeq o(n)$ or $o(n \log n)$ sols

Ranges < advanced ds: seg tree, bit, ...
may be stl? ad hoc?

Div2B? Don't overestimate.

let's try brute force

for every query

for every value in query

for every input range

sum

$= O(n^3)$ very slow

---

better brute force

step ①:

for every input range

accumulate in array

(1,7) (2,6) (3,10) (2,4) (6,8)

1	2	3	4	5	6	7	8	9	10
1	3	4	4	3	4	3	2	1	1

$O(n^2)$

e.g. 6 appears in 4 interval

step ②: change value to 1 if $v[i] \geq k$

for $k = 3$

0 1 1 1 1 1 1 0 0 0

$O(n^2)$

step ③ for given query

Just iterate & sum

$O(n^2)$

wait: sum of interval? prefix sum

so only $O(n)$

total $O(n^2) + O(n) = O(n^2)$

---

intended:

if we can solve step ① in

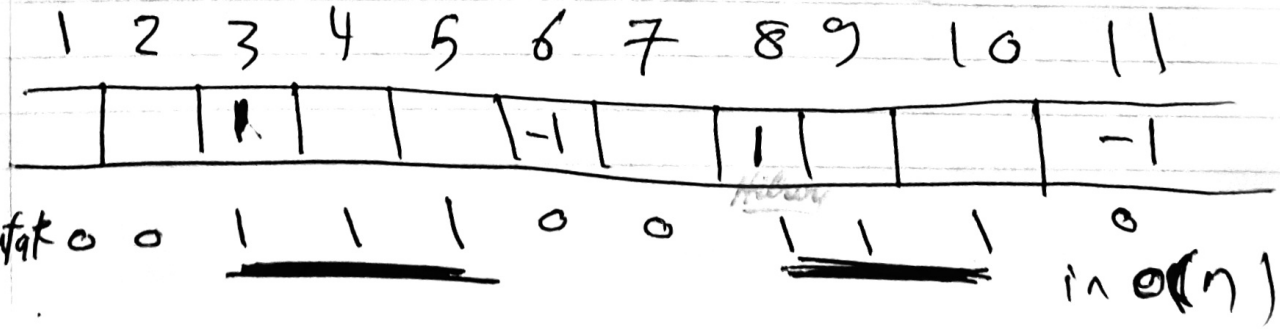
$O(n)$ = done

smart trick using prefix sum

say input range = { (3,5) (8,10) }

accumulate $a[s] + 1$ ~

$a[e+1] - 1$



in $O(n)$