J222 - Spicy Ramen

Ethen Yuen {ethening}

2022-01-29

香港電腦奧林匹克競賽
Hong Kong Olympiad in Informatics

Background

Problem Idea by ethening

Preparation by ethening, christycty

Problem Restatement

Given N + M inequalities

- $x \geq A1, x \geq A2, \ldots, x \geq AN$
- $x \leq B1, x \leq B2, \ldots, x \leq BM$

Find the number of integers in the range [0, R] that satisfies at least K inequalities.

2 2 3 70 30 50 40 65	27

4 2 6 50 25 20 0 10 50 40	16

3 3 4 100 41 71 89 0 23 53	0

Statistics

0 points	15 + 4 + 0 + 0 = 19
14 points	1 + 4 + 1 + 0 = 6
31 points	6 + 6 + 5 + 0 = 17
56 points	0 + 5 + 5 + 0 = 10
100 points	0 + 1 + 3 + 8 = 12

First solved by **cwong** at **23m 43s**

SUBTASK

For all cases:

$1 \le N, M \le 2 \times 10^5$

$1 < K \le N + M$

$1 < R \le 10^9$

$0 \le A_i, B_i \le R$

	Points	Constraints
1	14	$N = 2, M = 2, 3 \le K \le 4$ $1 \le R \le 3000$
2	17	$1 \le N, M \le 1000$ $1 \le R \le 3000$
3	25	$1 \le R \le 10^6$
4	17	$M = 2, K = N + M - 1$
5	27	No additional constraints

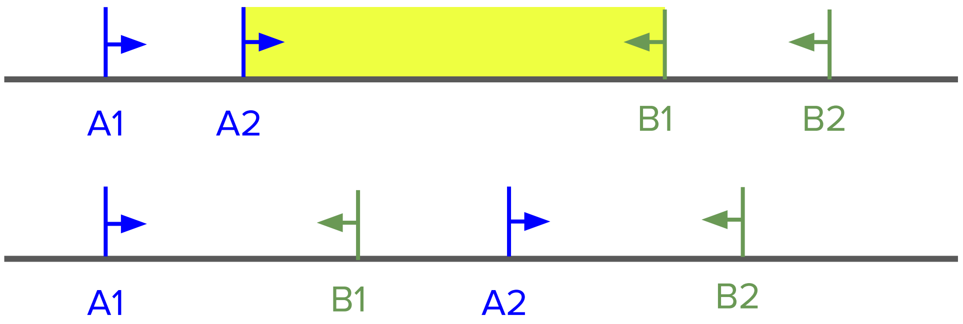
Subtask 1

Subtask 1 (14%): $N = M = 2$, $3 \le K \le 4$, $1 \le R \le 3000$.

- N, M, K are small enough to solve by doing some careful case handling.
- Suppose $A1 \le A2$ && $B1 \le B2$.

Subtask 1

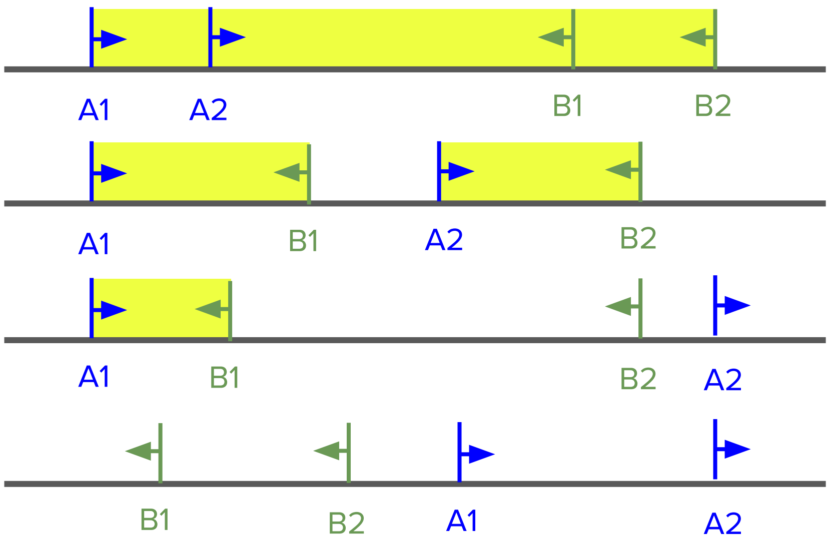
- Suppose A1 ≤ A2 && B1 ≤ B2. **K = 4**.



- If A2 ≤ B1, ans = B1 - A2 + 1; Else ans = 0.

Subtask 1

- Suppose A1 ≤ A2 && B1 ≤ B2. **K = 3**.



- If A2 ≤ B1, ans = B2 - A1 + 1;

  Else, separate handle

  [A1, B1] && [A2, B2].

Subtask 1

Subtask 1 (14%): $N = M = 2$, $3 \le K \le 4$, $1 \le R \le 3000$.

- N, M, K are small enough to solve by doing some careful case handling.
- After writing some ifs and doing some calculations...

Score: 14

Time Complexity: $O(1)$

Subtask 2

Subtask 2 (17%): $1 \le N, M \le 1000$, $1 \le R \le 3000$.

- N, M, R are small.
  - We could naively **try every x** in range [0, R], **test all N + M inequalities** against it.
  - Another way is to open **a counting array of [0, R]**. For each inequality, loop through value that satisfy it and **+1** to the counting array.
- Count those x that satisfy $\ge$ K inequalities.

Subtask 2

- Open **a counting array of [0, R]**. For each inequality, loop through value that satisfy it and **+1** to the counting array.

```
N=4 M=2 K=6 R=5
A={2 2 0 1}
B={5 4}
```

C[0]	C[1]	C[2]	C[3]	C[4]	C[5]
0	0	0	0	0	0
0	0	1	1	1	1
0	0	2	2	2	2

C[0]	C[1]	C[2]	C[3]	C[4]	C[5]
1	1	3	3	3	3
1	2	4	4	4	4
2	3	5	5	5	5
3	4	**6**	**6**	**6**	5

Subtask 2

Subtask 2 (17%): $1 \leq N, M \leq 1000$, $1 \leq R \leq 3000$.

- N, M, R are small.
  - We could naively **try every x** in range [0, R], **test all N + M inequalities** against it.
  - Another way is to open **a counting array of [0, R]**. For each inequality, loop through value that satisfy it and **+1** to the counting array.
- Count those x that satisfy ≥ K inequalities.

Score: 31

Time Complexity: O(R * (N + M))

Subtask 3

Subtask 3 (25%): $1 \leq R \leq 10^6$.

- Naively looping through value of x would lead to TLE.
- There exist faster way of update an array in range:
  - Optimize by using **Difference Array**.
  - Refer to the "Optimization and Common Tricks" lesson
- For increasing A[L..R] by 1,we could increase D[L] by 1 and decrease D[R + 1] by 1. At last, sum them up to get the final value.

Subtask 3

- Open **a differences array of [0, R + 1]**. For each inequality `l ≤ x ≤ r`, add 1 to D[L], subtract 1 to D[R + 1].

```
N=4 M=2 K=6 R=5
A={2 2 0 1}
B={5 4}
```

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]
0	0	0	0	0	0	0
0	0	1	0	0	0	-1
0	0	2	0	0	0	-2

D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]
1	0	2	0	0	0	-3
1	1	2	0	0	0	-4
2	1	2	0	0	0	-5
3	1	2	0	0	-1	-5

Subtask 3

Subtask 3 (25%): 1 ≤ R ≤ 10^6.

- At last, sum them up to get the final value.
- Count those x that C[x] ≥ K.

Score: 56

Time Complexity: O(N + M + R)

```
N=4 M=2 K=6 R=5
A={2 2 0 1}
B={5 4}
```

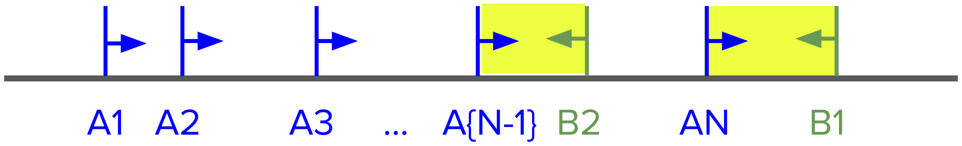
D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]
3	1	2	0	0	-1	-5

	C[0]	C[1]	C[2]	C[3]	C[4]	C[5]	
0	3	4	**6**	**6**	**6**	5	
+3	+1	+2	+0	+0	-1	-5	

Subtask 4

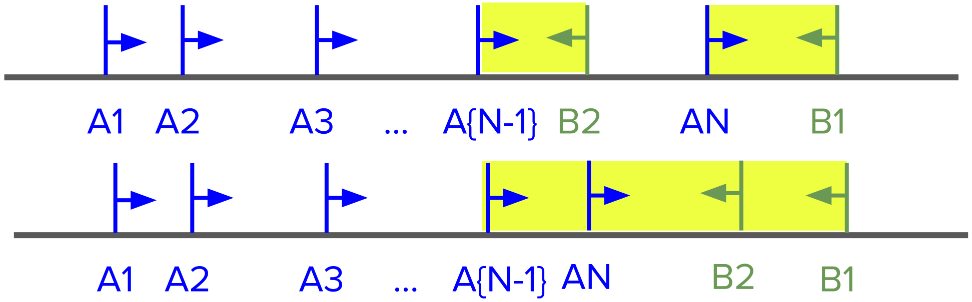
Subtask 4 (17%): M = 2, **K = N + M - 1**.

- A more complicated Subtask 1.
- The order of inequalities does not affect the result. Let's sort them for easier processing, i.e. A1 ≤ A2 ≤ … ≤ AN, B2 ≤ B1. (Notice that B is in reverse)



Subtask 4

- Subtask 4 (17%): M = 2, **K = N + M - 1**.



- If AN ≤ B2, ans = B1 - A{N-1} + 1; Else, separate handle [A{N-1}, B2] && [AN, B1].

Score: 17 (Cumulative: 73)

Time Complexity: O(N lg N + M lg M)

Full Solution

Subtask 5 (27%): No additional constraints.

- Let A be sorted in ascending order, B be sorted in descending order.
- Notice in Subtask 4, we group [A{N-2}, B2] && [AN, B1] together.
- For **x ≥ Ai**, if x is counted in the answer, **x ≤ B{K - i}.**
    - For convenience sake, we let **A0 be 0, and B0 be R**.
    - We could just do **Set Union** on **[A0, BK], [A1, B{K-1}], ..., [AN, B{K - N}]** (so that we won't double count overlapping interval) and the answer is the size of unioned set.

Full Solution

- We could just do **Set Union** on **[A0, BK], [A1, B{K-1}], ..., [AN, B{K-N}]** and the answer is the size of unioned set.
  - e.g. Union of [3, 10], [5, 14], [16, 17], [21, 25], [18, 26] is [3, 14], [16, 26].
  - Size of the union set is (14 - 3 + 1) + (26 - 16 + 1) = 23.
- The Set Union is less complicated because **A0 ≤ A1 ≤ ... ≤ AN**.
  - Just maintain the current interval right-bound and try to merge the next interval into it.
- Discard invalid interval where the right-bound < left-bound, also end the loop early if K-N becomes negative.

Full Solution

- We could just do **Set Union** on **[A0, BK], [A1, B{K-1}], ..., [A{min(K, N)}, B0]** and the answer is the size of unioned set.
- Alternatively, start with interval **[A0, A1 - 1], [A1, A2 - 1], ..., [A{min(K, N)}, R]**. Update first interval with ≤ **Bk**, second with ≤ **B{k - 1}**, and so on. (basically start with disjoint interval at first so no union is needed).

Score: 100

Time Complexity: O(N lg N + M lg M)