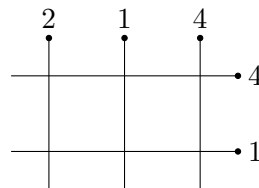


Political cost (cost)

In de stad waar je woont zijn er N straten die van oost naar west lopen (van straat 0 tot straat $N - 1$), en M lanen die van noord naar zuid lopen (van laan 0 tot laan $M - 1$). Elke straat en laan heeft een *Politiek belang*, hoe belangrijk de belangrijkste burger is die eraan woont. We geven de politieke belangen als twee arrays $A[1 \dots N]$ en $B[1 \dots M]$ met integers van 1 en met K . In het onderstaande diagram vind je zo'n stad met 2 straten en 3 lanen, met politieke belangen $A = [1, 4]$ en $B = [2, 1, 4]$.



De burgemeester wil een parade organiseren door de stad. Als de parade door de kruising van straat x en laan y loopt, zal het verkeer op beide wegen verstoord worden waarvoor de burgemeester een politiek verlies heeft van $\max(A[x], B[y])$. Als de parade door meerdere kruispunten gaat is het politieke verlies het *maximum* van het politieke verlies voor ieder kruispunt. Merk op dat de verliezen niet opgeteld worden; het maakt niet uit hoeveel mensen last hebben van de optocht, alleen de belangrijkste burger die wordt verstoord.

De *politieke afstand* tussen twee kruispunten is het kleinste *politieke verlies* van een parade die vertrekt vanaf het eerste kruispunt en aankomt op het tweede kruispunt. Jouw opdracht is om de som van alle politieke afstanden tussen alle paren kruispunten van de stad uit te rekenen.

Implementatie

Je moet één `.cpp`-bestand inleveren.

🔗 In de bijlagen van deze opgave vind je een sjabloon `cost.cpp` met een voorbeeld-implementatie.

Je moet de volgende functie implementeren:

```
C++ | int solve(int N, int M, int K, vector<int> A, vector<int> B);
```

- Integer N die staat voor het aantal straten van oost naar west.
- Integer M die staat voor het aantal lanen van noord naar zuid.
- De array A , genummerd van 0 tot en met $N - 1$, bevat de getallen A_0, A_1, \dots, A_{N-1} , waarbij A_i staat voor het politieke belang van de i de oost-west straat.
- De array B , genummerd van 0 tot en met $M - 1$, bevat de getallen B_0, B_1, \dots, B_{M-1} , waarbij B_i staat voor het politieke belang van de i de noord-zuid laan.
- De functie moet de som van alle politieke afstanden teruggeven, **modulo** 1000003.

De grader roept de functie `solve` aan en zal de teruggegeven waarde schrijven naar het uitvoerbestand.

Voorbeeld Grader

De map van de opdracht bevat een voorbeeldgrader, een versimpelde versie van de jury's grader, die je kan gebruiken om je oplossing lokaal te testen. De voorbeeldgrader leest de invoer van `stdin`, roept de functies aan die je moet implementeren, en schrijft vervolgens de uitvoer naar `stdout`.

De invoer bestaat uit 3 regels met:

- Regel 1: de integers N , M en K .
- Regel 2: de integers A_i , gescheiden door spaties.
- Regel 3: de integers B_i , gescheiden door spaties.

De uitvoer bestaat uit één regel die de waarde bevat die door de functie `solve` wordt teruggegeven.

Randvoorwaarden

- $1 \leq N \leq 3 \times 10^5$.
- $1 \leq M \leq 3 \times 10^5$.
- $1 \leq K \leq N + M$.
- $1 \leq A_i \leq K$ voor $i = 1 \dots N$.
- $1 \leq B_i \leq K$ voor $i = 1 \dots M$.

Scoring

Je programma zal worden getest op een set van testgevallen die per deelopgaven zijn gegroepeerd. Om de score van een deelopgave te krijgen moet je alle testgevallen daarvan goed oplossen.

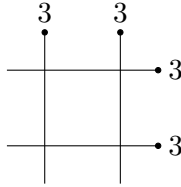
- **Subtask 1 [0 punten]**: Voorbeeld testgevallen.
- **Subtask 2 [10 punten]**: $N \leq 10^1, M \leq 10^1$.
- **Subtask 3 [10 punten]**: $N \leq 10^2, M \leq 10^2$.
- **Subtask 4 [10 punten]**: $N = 1, M \leq 10^4$.
- **Subtask 5 [10 punten]**: $N = 1, M \leq 10^5$.
- **Subtask 6 [10 punten]**: $N \leq 10^3, M \leq 10^3$.
- **Subtask 7 [10 punten]**: $N \leq 10^4, M \leq 10^4$.
- **Subtask 8 [10 punten]**: $N \leq 10^5, M \leq 10^5$ en de arrays A en B zijn niet-dalend, oftewel, als $i < j$, dan $A_i \leq A_j$ en $B_i \leq B_j$.
- **Subtask 9 [10 punten]**: $N \leq 10^5, M \leq 10^5, K \leq 10^1$.
- **Subtask 10[10 punten]**: $N \leq 10^5, M \leq 10^5$.
- **Subtask 11[10 punten]**: Geen extra randvoorwaarden, oftewel, $N \leq 3 \times 10^5, M \leq 3 \times 10^5$.

Voorbeelden

stdin	stdout
2 2 4 3 3 3 3	48
1 3 4 2 2 3 1	25
2 3 5 1 4 2 1 4	135

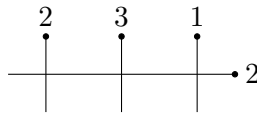
Uitleg

In het **eerste voorbeeld** hebben we een stad met 2 straten en 2 lanen die allemaal politiek belang 3 hebben:



Er zijn 16 verschillende paren kruispunten. Aangezien de politieke afstand tussen elk paar kruispunten 3 is, is de oplossing $3 \cdot 16 = 48$.

In het **tweede voorbeeld** zijn er 1 straat en 3 lanen, met politieke belangen $A = [2]$ en $B = [2, 3, 1]$:



Er zijn 9 paren kruispunten. Drie van deze paren starten en eindigen op hetzelfde kruispunt en hebben politieke afstanden van 2, 3 en 2 respectievelijk (de 3e laan heeft een politiek belang van 1, maar het politieke belang van de 1e is 2, dus de politieke afstand van een optocht die vertrekt en aankomt op het kruispunt tussen de 3e laan en 1e straat is minstens 2). Elk ander paar van kruispunten kruist de 2e laan, en moet dus een politieke afstand van 3 hebben. Dus de totale som is $2 + 3 + 2 + 6 \cdot 3 = 25$.

Het **derde voorbeeld** komt overeen met het voorbeeld dat gegeven word in de uitleg van de opdracht. Hier zijn er 2 straten en 3 lanen. Je kan controleren, met wat geduld, dat de som van politieke afstanden van de 36 paren kruispunten 135 is.