

ESI - Double Partition

Tout le monde sait que Red1 fait son diplôme d'ingénieur à l'ESI-SBA (Ecole Supérieure en Informatique, Sidi-Bel-Abbès).

Red1 se distingue par son excellence en Algorithmique, et s'avère le premier choix de l'université pour directeur du nouveau projet collaboratif "EFFICIENT SEGMENT INTERCHANGE DUAL PARTITIONING", le but étant d'accélérer le transfert entre les différentes bases de données de l'école.

Son Excellence Red1 doit gérer n et m packets de données provenant de la base de données des élèves et celle des professeurs, respectivement. La taille du i ème packet de la première pile est donné par a_i , la taille du j ème packet de la deuxième pile est donné par b_j . Leurs tailles sont données en Giga octets (Go). Red1 veut déplacer tous ces packets vers une base de données centralisée.

Malheureusement, il y a des restrictions sur la manière dont il peut procéder: pour chaque transfert, Red1 peut prendre un nombre de packets (potentiellement 0) par pile, mais il doit toujours les transférer en ordre, c'est à dire que s'il prend un packet i de la pile 1, cela signifie qu'il a déjà transféré le packet $i + 1$, ou qu'il va être transféré à ce moment (même raisonnement pour la pile 2). De plus, il ne peut envoyer qu'une taille totale de w Go par transfert. Chaque requête prend k secondes quelque soit le volume des données envoyées.

Avant de coder le processus de transfert, Red1 veut une estimation du temps minimal qu'il devra prendre pour compléter sa tâche sur deux bases de données connues. Aidez Red1 en calculant ce temps pour chaque paire de bases de données fournies.

Contraintes

- $1 \leq n, m \leq 2000$
- $1 \leq a_i, b_j \leq w \leq 10^9$
- $1 \leq k \leq 100$

I/O

Soit t^* le temps optimal pour compléter la tâche.

Entrée

```
n m w k
a1 a2 ... a_n
b1 b2 ... b_m
```

Sortie

```
t*
```

Sous-tâches

La note finale de cette tâche est donnée par la somme des points des sous-tâches que vous avez réussi dans au moins une de vos soumissions.

Test group	Points	Constraints
1	5	$a_1 = a_2 = \dots = w$
2	5	$a_1 = a_2 = \dots = a_n = b_1 = b_2 = \dots = b_m$
3	5	$a_1 = a_2 = \dots = a_n = 1$
4	5	$n, m \leq 7$
5	20	$n, m \leq 100$
6	10	$n, m \leq 200$
7	20	$n, m \leq 500$
8	30	Aucune contrainte additionnelle.

Exemple

Entrée

```
4 5 11 1
4 3 7 5
3 4 3 6 2
```

Sortie

```
4
```

Explication

Le temps minimal pour accomplir cette tâche est de 4 secondes, de la manière suivante:

- Au premier transfert, On prend: les packets a_4 et b_5 ayant une taille totale de 7, ce qui est possible puisque la taille est inférieure à 11.
- Ensuite, on transfert a_3 et a_2 ayant tous deux une taille totale de 10.
- Au troisième transfert, on prend les packets a_1 et b_4 ayant une taille totale de 10.
- Finalement, on envoie les packets de taille b_3 , b_2 and b_1 de taille totale 10.