



Livres Anciens (Ancient Books)

La ville de Téhéran abrite la bibliothèque nationale d'Iran. Le joyau de cette bibliothèque est situé dans une longue salle, sur une rangée de n tables numérotées de 0 à $n - 1$ (de gauche à droite). Sur chaque table est exposé un manuscrit ancien. Ces livres sont triés par ordre chronologique, ce qui rend difficile pour les visiteurs la recherche d'un livre à partir de son titre. Le gestionnaire de la bibliothèque a donc décidé de trier les livres dans l'ordre alphabétique de leur titre.

Aryan, un bibliothécaire, a hérité de cette tâche. Il a créé une liste p de longueur n , contenant tous les nombres entiers entre 0 et $n - 1$. Cette liste décrit les changements nécessaires pour réorganiser les livres par ordre alphabétique : pour tout $0 \leq i \leq n$, le livre initialement sur la table i doit être déplacé vers la table $p[i]$.

Aryan commence à trier les livres à la table s . Il veut revenir à cette même table après avoir réorganisé tous les livres. Comme les livres sont très précieux, il ne peut en transporter qu'un seul à la fois. Lors du tri des livres, Aryan va effectuer une série d'actions. Chacune d'entre elles sera de l'un des types suivants :

- S'il ne transporte pas de livre et qu'il y a un livre sur la table devant lui, il peut prendre ce livre.
- S'il transporte un livre et qu'il y a un autre livre sur la table devant lui, il peut échanger son livre avec celui sur la table.
- S'il transporte un livre et qu'il est devant une table vide, il peut déposer le livre sur la table.
- Il peut marcher jusqu'à la table de son choix. Il peut en profiter pour transporter un livre.

Pour tout $0 \leq i, j \leq n - 1$, la distance entre les tables i et j est exactement $|i - j|$ mètres. Votre tâche est d'aider Aryan à trier les livres de manière à ce que la distance totale parcourue soit minimisée.

Détails d'implémentation

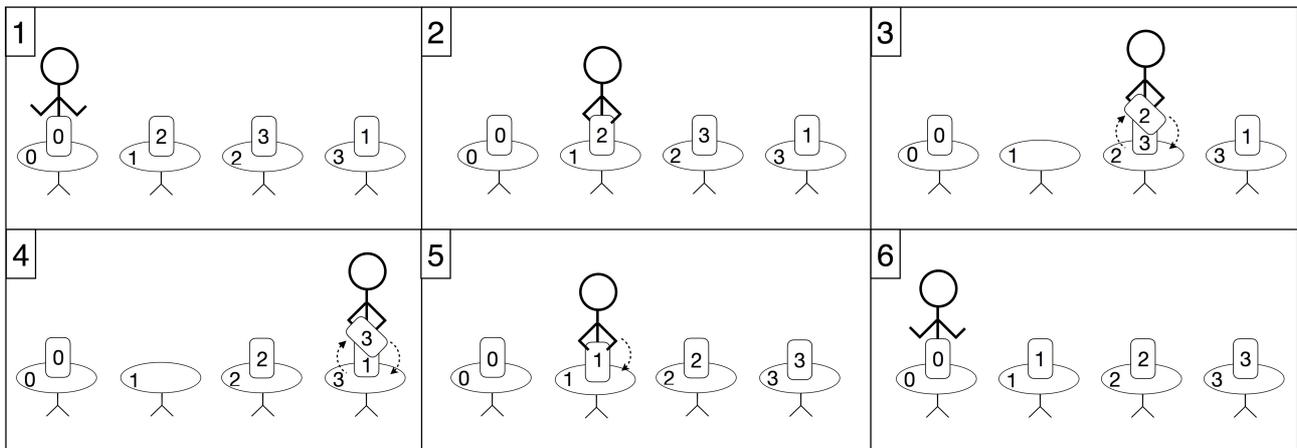
Vous devez implémenter la fonction suivante :

```
int64 minimum_walk(int[] p, int s)
```

- p est un tableau de taille n . Le livre initialement sur la table i doit être déposé par Aryan sur la table $p[i]$ (pour tout $0 \leq i < n$).
- s est le numéro de la table devant laquelle est Aryan au début, et où il doit être après avoir trié les livres.
- Cette fonction doit renvoyer la distance totale minimale (en mètres) qu'Aryan doit parcourir afin de trier les livres.

Exemple

```
minimum_walk([0, 2, 3, 1], 0)
```



Dans cet exemple, $n = 4$ et Aryan commence à la table 0. Il trie les livres de la manière suivante :

- Il marche jusqu'à la table 1 et prend le livre qui y est exposé. Ce livre doit être déposé sur la table 2.
- Ensuite, il marche jusqu'à la table 2 et échange son livre avec celui présent sur la table. Le nouveau livre qu'il transporte doit être déposé sur la table 3.
- Ensuite, il marche jusqu'à la table 3 et échange son livre avec celui présent sur la table. Le nouveau livre qu'il transporte doit être déposé sur la table 1.
- Ensuite, il marche jusqu'à la table 1 et dépose le livre qu'il transporte.
- Enfin, il retourne à table 0.

Notez que le livre sur la table 0 est déjà au bon endroit, donc Aryan n'a pas à le déplacer. Dans cette solution, la distance totale parcourue est de 6 mètres. C'est la solution optimale ; par conséquent la fonction doit renvoyer 6.

Contraintes

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- Le tableau p contient n entiers distincts entre 0 et $n - 1$ (inclus).

Sous-tâches

1. (12 points) $n \leq 4$ et $s = 0$
2. (10 points) $n \leq 1000$ et $s = 0$
3. (28 points) $s = 0$
4. (20 points) $n \leq 1000$
5. (30 points) Aucune contrainte supplémentaire.

Évaluateur d'exemple

L'évaluateur d'exemple lit les entrées dans le format suivant :

- ligne 1: n s
- ligne 2: $p[0]$ $p[1]$... $p[n - 1]$

L'évaluateur d'exemple écrit une seule ligne sur la sortie standard, qui contient la valeur renvoyée par la fonction `minimum_walk`.