



## Tomi antichi

A Teheran, nella Biblioteca Nazionale Iraniana, si trova una lunga sala con una fila di  $n$  tavoli (numerati da 0 a  $n - 1$ ) contenenti ciascuno un antico tomo manoscritto in esposizione. I tomi al momento sono ordinati per età, a tutto svantaggio dei visitatori che li cercano solitamente in base al titolo. Quindi, il direttore della biblioteca ha deciso di riordinare i tomi in ordine alfabetico di titolo.

Aryan, il bibliotecario incaricato per questo lavoro, ha preparato una lista  $p$  di lunghezza  $n$  contenente interi **distinti** da 0 a  $n - 1$ . La lista descrive i cambiamenti che è necessario apportare: per ogni  $0 \leq i < n$ , il tomo correntemente sul tavolo  $i$  deve essere spostato sul tavolo  $p[i]$ .

La postazione di Aryan si trova al tavolo  $s$ , da cui deve partire e ritornare dopo aver finito il lavoro. Dato che i tomi sono molto preziosi, Aryan non può trasportare più di un tomo per volta: infatti le azioni che può fare sono esattamente le seguenti:

- Se non ha un tomo in mano, può prendere il tomo eventualmente presente nel tavolo in cui si trova.
- Se ha un tomo in mano, può scambiarlo con il tomo eventualmente presente nel tavolo in cui si trova.
- Se ha un tomo in mano e non ci sono tomi nel tavolo corrente, può lasciarlo su quel tavolo.
- Può camminare verso un qualunque tavolo, trasportando *al massimo* un singolo tomo nel percorso.

Sapendo che la distanza tra i tavoli  $i$  e  $j$  è pari a  $|j - i|$  metri (per ogni  $0 \leq i, j \leq n - 1$ ), aiuta Aryan a riordinare i tomi camminando il meno possibile.

## Dettagli di implementazione

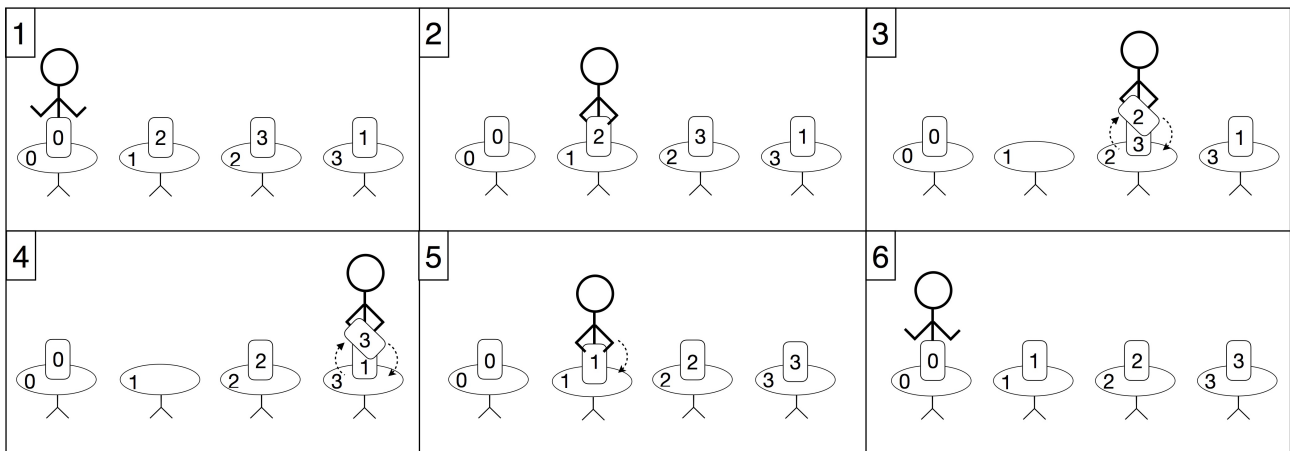
Devi implementare la seguente funzione:

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

- $p$ : un array di lunghezza  $n$ , per cui il tomo sul tavolo  $i$  deve essere spostato sul tavolo  $p[i]$  (per ogni  $0 \leq i < n$ ).
- $s$ : il tavolo in cui si trova Aryan all'inizio e al quale deve tornare alla fine.
- La funzione deve restituire la minima distanza totale (in metri) che Aryan deve percorrere per riordinare i tomi.

## Esempio

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



In questo esempio,  $n = 4$  e la postazione di Aryan è al tavolo 0. Aryan riordina i tomi come segue:

- Cammina fino al tavolo 1 e ne raccoglie il tomo corrispondente (che deve essere spostato al tavolo 2).
- Cammina fino al tavolo 2 e scambia il tomo che sta portando (che è ora a posto) con quello presente (che deve essere spostato al tavolo 3).
- Cammina fino al tavolo 3 e scambia il tomo che sta portando (che è ora a posto) con quello presente (che deve essere spostato al tavolo 1).
- Cammina fino al tavolo 1 e posa il tomo che sta portando sul tavolo.
- Ritorna al tavolo 0.

Nota che il tomo sul tavolo 0 è già nel posto corretto, quindi Aryan non ha bisogno di raccoglierlo. La distanza totale percorsa in questa soluzione è di 6 metri, che è ottimale, quindi la funzione deve restituire 6.

## Assunzioni

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- L'array  $p$  contiene  $n$  interi distinti tra 0 e  $n - 1$  (estremi inclusi).

## Assegnazione del punteggio

1. **(12 punti)**  $n \leq 4$ ,  $s = 0$
2. **(10 punti)**  $n \leq 1000$ ,  $s = 0$
3. **(28 punti)**  $s = 0$
4. **(20 punti)**  $n \leq 1000$
5. **(30 punti)** Nessuna limitazione specifica.

## Grader di prova

Il grader di prova legge l'input nel seguente formato:

- riga 1:  $n$   $s$
- riga 2:  $p[0]$   $p[1]$   $\dots$   $p[n - 1]$

Il grader di prova stampa una sola riga contenente il valore restituito da `minimum_walk`.