



## Libros antiguos

La ciudad de Teherán es hogar de la Biblioteca Nacional de Irán. El tesoro principal de esta biblioteca está ubicado en un gran salón con una fila de  $n$  mesas, numeradas del 0 al  $n - 1$  de izquierda a derecha. En cada mesa hay un libro manuscrito antiguo en exposición. Estos libros están ordenados por antigüedad, lo cual dificulta para los visitantes buscar libros por su título. Por esta razón, la administradora de la biblioteca ha decidido ordenar los libros alfabéticamente por sus títulos.

Aryan, un bibliotecario, es quien hará el trabajo. Él ha creado una lista  $p$  de longitud  $n$ , que contiene enteros diferentes desde 0 hasta  $n - 1$ . Esta lista describe los cambios necesarios para reacomodar los libros en orden alfabético: para cada  $0 \leq i < n$ , el libro que está actualmente en la mesa  $i$  debe ser movido a la mesa  $p[i]$ .

Aryan empieza a ordenar los libros en la mesa  $s$ . Él quiere volver a la misma mesa después de terminar el trabajo. Como los libros son muy valiosos, él no puede cargar más de uno a la vez. Mientras ordena los libros, Aryan ejecutará una secuencia de acciones. Cada una de ellas debe ser una de las siguientes posibles acciones:

- Si no está cargando un libro y hay un libro sobre la mesa en la que él está, puede recoger este libro.
- Si está cargando un libro y hay otro libro sobre la mesa en la que él está, puede intercambiar ambos libros.
- Si está cargando un libro y él está en una mesa vacía, puede poner el libro que está cargando sobre la mesa.
- Él puede caminar a cualquier mesa. Él puede cargar un libro mientras camina.

Para cada  $0 \leq i, j \leq n - 1$ , la distancia entre las mesas  $i$  y  $j$  es exactamente  $|j - i|$  metros. Tu tarea es calcular la mínima distancia total que Aryan debe caminar para ordenar todos los libros.

## Detalles de implementación

Debes implementar la siguiente función:

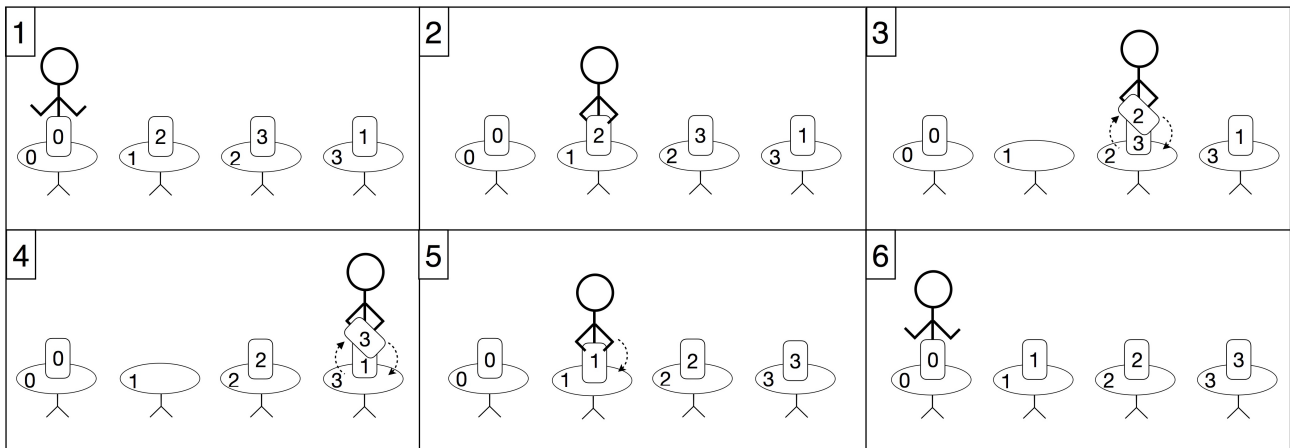
```
int64 minimum_walk(int[] p, int s)
```

- $p$  es un arreglo de longitud  $n$ . El libro que inicialmente está en la mesa  $i$  debe ser llevado por Aryan a la mesa  $p[i]$  (para todo  $0 \leq i < n$ ).
- $s$  es el número de la mesa donde Aryan comienza, y donde debe estar después de ordenar todos los libros.

- Esta función debe retornar la mínima distancia total (en metros) que Aryan debe caminar para ordenar los libros.

## Ejemplo

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



En este ejemplo,  $n = 4$  y Aryan empieza en la mesa 0. Él ordena los libros de la siguiente forma:

- Él camina a la mesa 1 y recoge el libro que hay sobre ella. Este libro debe ser puesto sobre la mesa 2.
- A continuación, él camina a la mesa 2 y cambia el libro que está cargando con el que está sobre la mesa. El nuevo libro que está cargando debe ser llevado a la mesa 3.
- Después, él camina a la mesa 3 y cambia el libro que está cargando con el libro sobre la mesa. El nuevo libro que está cargando debe ser puesto en la mesa 1.
- Luego, él camina a la mesa 1 y coloca el libro que está cargando sobre la mesa.
- Finalmente, él camina de vuelta a la mesa 0.

Nota que el libro en la mesa 0 ya está en el lugar correcto, la mesa 0, así que Aryan no lo tiene que recoger. La distancia total que él camina en esta solución es 6 metros. Esta es la solución óptima; por lo tanto, el procedimiento debe retornar 6.

## Restricciones

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- El arreglo  $p$  contiene  $n$  enteros diferentes entre 0 y  $n - 1$ , inclusive.

## Subtareas

1. (12 puntos)  $n \leq 4$  y  $s = 0$
2. (10 puntos)  $n \leq 1000$  y  $s = 0$

3. (28 puntos)  $s = 0$
4. (20 puntos)  $n \leq 1000$
5. (30 puntos) no hay restricciones adicionales

## Calificador de ejemplo

El calificador de ejemplo lee la entrada en el siguiente formato:

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

El calificador de ejemplo imprime una sola línea que contiene el valor retornado por `minimum_walk`.