



## Libros antiguos

La ciudad de Teheran es hogar de la Biblioteca Nacional de Irán. El tesoro principal de esta biblioteca está ubicado en una larga sala con una fila de  $n$  mesas, enumeradas del 0 al  $n - 1$  de izquierda a derecha. En cada mesa hay un libro antiguo manuscrito en exposición. Estos libros están ordenados por antigüedad, lo cual hace difícil para los visitantes buscar libros por título. Así que, la administradora de la biblioteca ha decidido ordenar los libros alfabéticamente por sus títulos.

Aryan, un bibliotecario, va a hacer el trabajo. Él ha creado una lista  $p$  de longitud  $n$ , que contiene todos los 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 estas acciones debe ser una de las siguientes:

- Si él no está cargando un libro y hay un libro sobre la mesa en la que él está, él puede recoger este libro.
- Si él está cargando un libro y hay otro libro sobre la mesa en la que él está, él puede cambiar el libro que está cargando con el que está sobre la mesa.
- Si él está cargando un libro y él está en una mesa vacía, él 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. Su tarea es calcular la mínima distancia total que Aryan debe caminar para ordenar los libros.

## Detalles de implementación

Debe implementar el siguiente procedimiento:

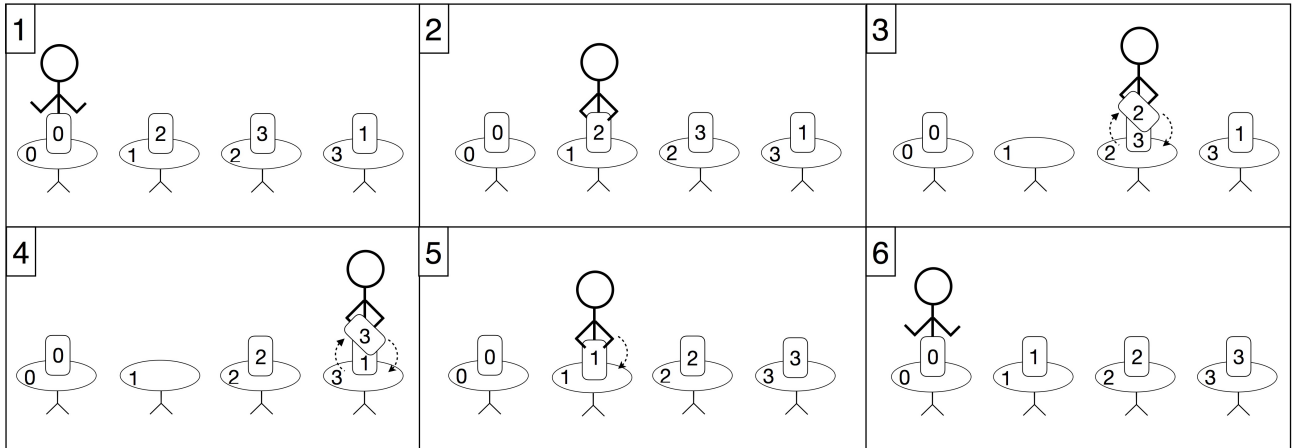
```
int64 minimum_walk(int[] p, int s)
```

- $p$  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.
- Este procedimiento 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 así:

- Él camina a la mesa 1 y recoge el libro que hay sobre ella. Este libro debe ser puesto sobre la mesa 2.
- Después, 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.
- Después, 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.

Note 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`.