



Libros Antiguos

La ciudad de Teherán es el hogar de la Biblioteca Nacional de Irán. El tesoro principal de esta biblioteca se encuentra en un salón largo con una fila de n mesas, enumeradas de 0 a $n - 1$ de izquierda a derecha. En cada mesa se expone un antiguo libro manuscrito. Estos libros se ordenan en función de su antigüedad, lo que hace que sea difícil para los visitantes buscar libros por el título. Por lo tanto, el administrador de la biblioteca ha decidido ordenar los libros en el orden alfabético de sus títulos.

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

Aryan comienza a ordenar los libros en la mesa s . Él quiere volver a la misma mesa después de terminar el trabajo. Dado que los libros son muy valiosos, no puede llevar más de un libro a la vez. Mientras ordena los libros, Aryan realizará una secuencia de acciones. Cada una de esas acciones debe ser una de las siguientes:

- Si no lleva un libro y hay un libro sobre la mesa en la que él está, puede recoger este libro.
- Si está llevando un libro y hay otro libro sobre la mesa en la que él está, puede cambiar el libro que lleva con el que está sobre la mesa.
- Si está llevando un libro y él está en una mesa vacía, puede poner el libro que lleva sobre la mesa.
- Puede caminar a cualquier mesa. Puede llevar un solo libro mientras camina.

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

Detalles de implementación

Debes implementar la siguiente función:

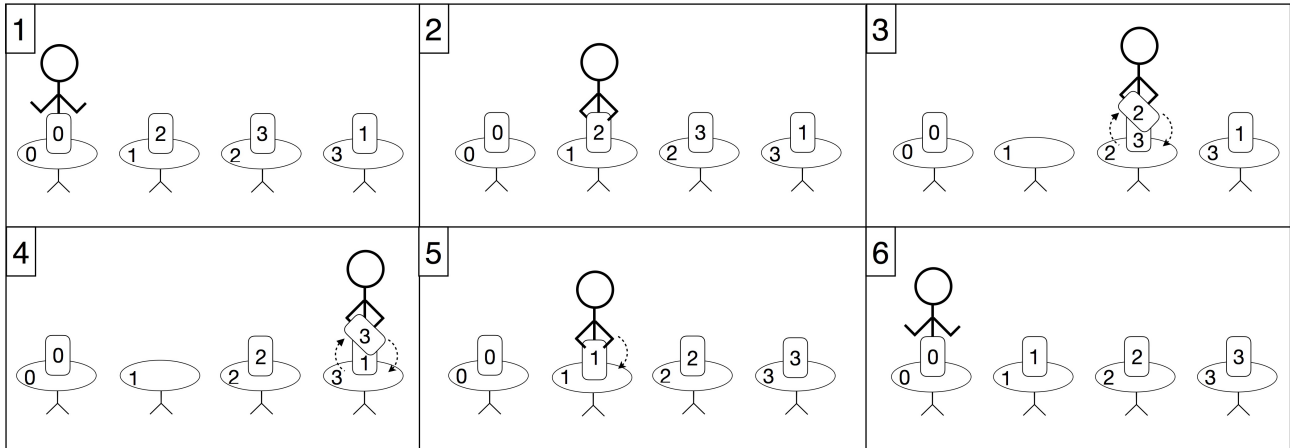
```
int64 minimum_walk(int[] p, int s)
```

- p es un arreglo de longitud n . El libro que se encuentra inicialmente 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 está al inicio, y donde debe estar después de ordenar los libros.
- Esta función debe devolver la mínima distancia total (en metros) que Aryan tiene que caminar

para ordenar los libros.

Ejemplos

```
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 manera:

- Camina a la mesa 1 y recoge el libro que hay sobre ella. Este libro debe ponerse en la mesa 2.
- Luego, camina a la mesa 2 y cambia el libro que lleva con el libro sobre la mesa. El nuevo libro que lleva debe ser puesto en la mesa 3.
- Luego, camina a la mesa 3 y cambia el libro que lleva con el libro sobre la mesa. El nuevo libro que está llevando debe ponerse en la mesa 1.
- Luego, camina a la mesa 1 y pone el libro que está llevando sobre la mesa.
- Finalmente, regresa a la mesa 0.

Tener en cuenta que el libro en la mesa 0 ya está en el lugar correcto, la mesa 0, por lo que Aryan no tiene que recogerlo. La distancia total que camina en esta solución es 6 metros. Esta es la solución óptima; por lo tanto, la función debe devolver 6.

Restricciones

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- El arreglo p contiene n enteros distintos 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) sin restricciones adicionales

Evaluador de ejemplo

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

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

El evaluador de ejemplo imprime una sola línea que contiene el valor devuelto de `minimum_walk`.