



Ancient Books

A cidade de Teerão é a casa da Biblioteca Nacional do Irão. O principal tesouro da biblioteca está localizado num longo corredor com um linha de n mesas, numeradas de 0 a $n - 1$ da esquerda para direita. Em cada mesa está exposto um livro antigo manuscrito. Estes livros estão ordenados de acordo com a sua idade, o que dificulta a tarefa dos visitantes que querem procurar um livro pelo seu título. Então, o bibliotecário decidiu ordenar os livros por ordem alfabética dos seus títulos.

Aryan, um dos funcionários, vai fazer este trabalho. Ele criou uma lista p de tamanho n contendo diferentes inteiros de 0 a $n - 1$. Esta lista descreve as mudanças necessárias para colocar os livros por ordem alfabética: para qualquer $0 \leq i < n$, o livro que está presentemente na mesa i deve ser movido para a mesa $p[i]$.

Aryan começa a sua tarefa de ordenar os livros na mesa s . Ele quer regressar à mesma mesa depois de terminar o seu trabalho. Como os livros são muito valiosos, ele não pode transportar mais do que um livro de cada vez. Para ordenar os livros, Aryan irá fazer uma sequência de ações. Cada uma destas ações será uma das seguintes:

- Se não estiver a transportar um livro e existir um livro na mesa onde ele está, pode decidir pegar o livro.
- Se estiver a transportar um livro e existir um outro livro na mesa onde ele está, pode decidir trocar o livro que está a transportar com o livro que está na mesa.
- Se estiver a transportar um livro e estiver numa mesa livre, pode decidir colocar o livro que transporta na mesa.
- Ele pode andar para qualquer mesa, mas pode transportar apenas um único livro enquanto anda.

Para qualquer $0 \leq i, j \leq n - 1$, a distância entre as mesas i e j é de $|j - i|$ metros. A sua tarefa é calcular a distância mínima total que Aryan precisa de caminhar para ordenar todos os livros.

Detalhes de implementação

Deve implementar a seguinte função:

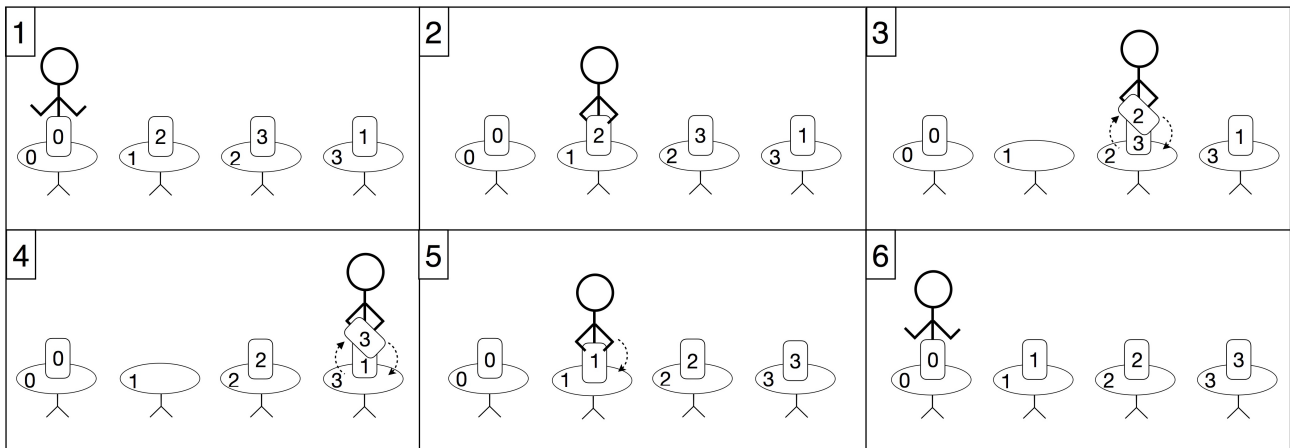
```
int64 minimum_walk(int[] p, int s)
```

- p é um vetor de tamanho n . O livro que está na mesa i no início deve ser transportado pelo Aryan para a mesa $p[i]$ (para qualquer $0 \leq i < n$).
- s é o número da mesa onde o Aryan está posicionado no início, e onde ele deve estar depois de ordenar os livros.

- Esta função deve retornar a mínima distância total (em metros) que Aryan tem de andar de forma a ordenar os livros.

Exemplo

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



Neste exemplo, $n = 4$ e Aryan está na mesa 0 no início. Ele ordena os livros da seguinte maneira:

- Ele anda para a mesa 1 e pega no livro dessa mesa. Este livro deve ser colocado na mesa 2.
- Depois, ele anda para a mesa 2 e troca o livro que transporta com o livro na mesa. O novo livro que agora transporta deve ser colocado na mesa 3.
- Depois, ele anda para a mesa 3 e troca o livro que transporta com o livro na mesa. O novo livro que agora transporta deve ser colocado na mesa 1.
- Depois ele anda para a mesa 1 e coloca o livro que transporta na mesa.
- Finalmente, ele anda de volta para a mesa 0.

Note que o livro da mesa 0 já está colocado no sítio correto, a mesa 0, e desse modo Aryan não precisa de pegar nele. Este é o procedimento ótimo e por isso mesmo a função deve retornar 6.

Restrições

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- O vetor p contém n inteiros distintos entre 0 e $n - 1$, inclusive.

Subtarefas

1. (12 pontos) $n \leq 4$ e $s = 0$
2. (10 pontos) $n \leq 1000$ e $s = 0$
3. (28 pontos) $s = 0$
4. (20 pontos) $n \leq 1000$

5. (30 pontos) nenhuma restrição adicional

Avaliador de exemplo

O avaliador de exemplo lê o *input* no seguinte formato:

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

O avaliador exemplo escreve uma única linha contendo o valor de retorno de `minimum_walk`.