



Ancient Books

Grad Teheran je dom nacionalne knjižnice Irana. Glavno se blago knjižnice nalazi u dugačkom redu s n stolova koji su označeni od 0 do $n - 1$ slijeva nadesno. Na svakom se stolu nalazi jedna rukom ispisana knjiga. Te su knjige poredane po starosti pa je posjetiteljima teško pronaći knjigu koju traže ako znaju samo njen naslov. Shodno tome, upravitelj knjižnice Krešimir Malnar, želi poredati knjige prema abecednom poretku njihovih naslova.

Požidar Stjepanović je dežurni knjižničar koji će odraditi posao. Napravio je listu p duljine n koja sadrži različite cijele brojeve između 0 i $n - 1$. Ta lista opisuje promjene koje je potrebno napraviti kako bi posao bio uspješno obavljen: za sve $0 \leq i < n$, knjiga koja se trenutno nalazi na stolu i treba se premjestiti na stol $p[i]$.

Požidar započinje sa sortiranjem knjiga na stolu s . Nakon obavljenog posla želi se vratiti na taj isti stol. Budući da su knjige jako vrijedne i da u jednoj ruci drži bocu, ne može nositi više od jedne knjige istovremeno. Prilikom sortiranja, Požidar će napraviti niz radnji koje su opisane u nastavku:

- ako ne nosi knjigu, a knjiga se nalazi na stolu kod kojeg se nalazi, Požidar ju smije uzeti.
- Ako nosi knjigu i neka se druga knjiga nalazi na stolu, Požidar ih može zamijeniti.
- Ako nosi knjigu i naišao je na prazan stol, Požidar može knjigu ostaviti tamo.
- Požidar se može slobodno kretati do bilo kojeg stola noseći najviše jednu knjigu i neograničen broj boca.

Za sve $0 \leq i, j \leq n - 1$, udaljenost između stolova i i j je točno $|j - i|$ pedalja. Vaš je zadatak pomoći Požidaru da sortira knjige tako da je ukupna prohodana udaljenost (u pedljima) minimizirana.

Implementacijski detalji

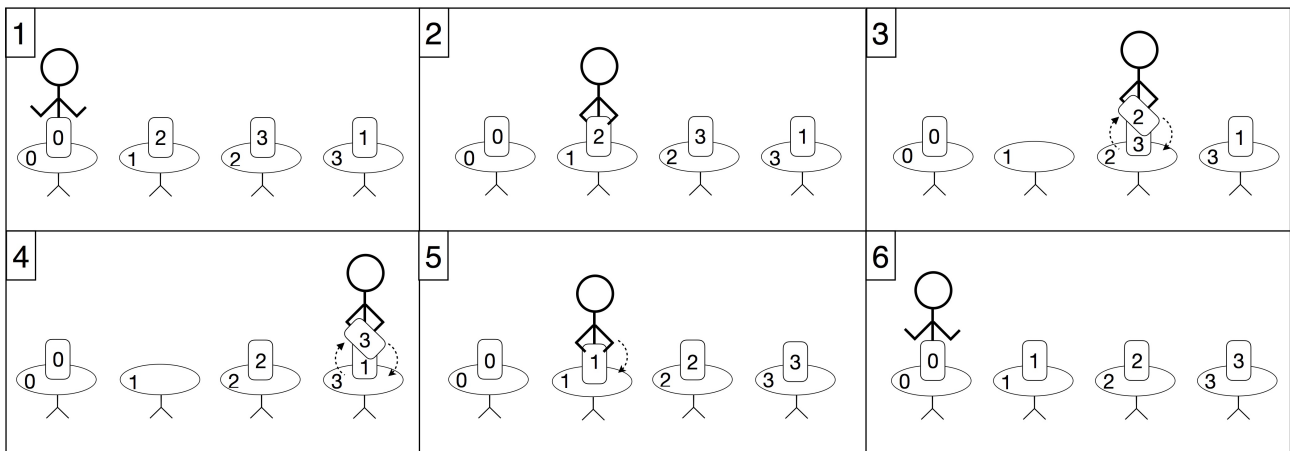
Trebate implementirati sljedeću funkciju:

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

- p je niz duljine n . Knjiga koja se na početku nalazi na stolu i treba biti odnešena (od strane Požidara) do stola $p[i]$ (za sve $0 \leq i < n$).
- s je oznaka stola gdje se Požidar inicijalno nalazi i gdje se mora vratiti nakon sortiranja knjiga (po novu bocu).
- Funkcija treba vratiti najmanju udaljenost (u pedljima) koju Požidar mora prohodati kako bi sortirao knjige.

Primjer

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



U ovom primjeru, $n = 4$ i Požidar se inicijalno nalazi na stolu 0. Sortira knjige na sljedeći način:

- Hoda do stola 1 i pokuplja knjigu koja se tamo nalazi. Ta se knjiga treba nalaziti na stolu 2.
- Nakon toga, hoda do stola 2 i mijenja knjigu koju nosi za onu koja se nalazi na tom stolu. Nova knjiga koju nosi treba se nalaziti na stolu 3.
- Zatim, hoda do stola 3 i mijenja knjigu koju nosi za onu koja se nalazi na tom stolu. Nova knjiga koju nosi treba se nalaziti na stolu 1.
- Potom, hoda do stola 1 i tamo stavlja knjigu koju je nosio u ruci.
- Konačno, vraća se do stola 0 po novu bocu.

Primijetite da se knjiga na stolu 0 već nalazi na ispravnom mjestu pa ju Požidar ne mora pokupiti. Ukupna udaljenost koju je prošao iznosi 6 pedalja. To je i optimalno rješenje, stoga, funkcija treba vratiti 6.

Ograničenja

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- Niz p sastoji se od n različitih cijelih brojeva između 0 i $n - 1$, uključivo.

Podzadaci

1. (12 bodova) $n \leq 4$ i $s = 0$
2. (10 bodova) $n \leq 1000$ i $s = 0$
3. (28 bodova) $s = 0$
4. (20 bodova) $n \leq 1000$
5. (30 bodova) bez dodatnih ograničenja

Sempl grejder

Sempl grejder čita ulaz u sljedećem formatu:

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

Sempl grejder ispisuje jednu liniju koja sadrži vrijednost `minimum_walk`.