



Stare książki

W Teheranie znajduje się Narodowa Biblioteka Iranu. Najcenniejszy zbiór tej biblioteki znajduje się w długim korytarzu z rzędem n stołów, oznaczonych kolejnymi liczbami naturalnymi od 0 do $n - 1$ włącznie, od lewej do prawej. Na każdym stole leży dokładnie jedna książka ze starożytnym rękopisem. Książki te są obecnie posortowane według ich wieku, co utrudnia czytelnikom wyszukiwanie książek według tytułu. Dyrektor biblioteki postanowił więc uporządkować książki według tytułów (alfabetycznie).

Bibliotekarz Aryan ma wykonać to zadanie. Stworzył listę p długości n , zawierającą różne liczby naturalne od 0 do $n - 1$. Lista opisuje zmiany niezbędne do uporządkowania książek w kolejności alfabetycznej (tytułów): dla każdego $0 \leq i < n$, książka obecnie znajdująca się na stole i powinna być położona na stole $p[i]$.

Aryan rozpoczyna sortowanie książek przy stole s . Chce powrócić do tego samego stołu po zakończeniu pracy. Ponieważ książki są bardzo cenne, nie może on nieść więcej niż jednej książki na raz. Podczas sortowania książek Aryan wykona ciąg akcji. Każda z nich będzie jedną z następujących:

- Jeśli obecnie nie trzyma książki i przy stole, przy którym stoi, jest książka, może ją podnieść.
- Jeśli trzyma książkę i przy stole, przy którym stoi, znajduje się inna książka, może zamienić trzymaną książkę z tą na stole.
- Jeśli trzyma książkę i znajduje się przy pustym stole, może położyć książkę na stół.
- Może podejść do dowolnego stołu. Podczas tego może (ale nie musi) nieść pojedynczą książkę.

Dla każdego $0 \leq i, j \leq n - 1$ odległość pomiędzy stołami i oraz j wynosi dokładnie $|j - i|$ metrów. Twoim zadaniem jest pomóc Aryanowi posortować książki w taki sposób, żeby łączny pokonany dystans był najmniejszy możliwy.

Szczegóły implementacyjne

Należy zaimplementować funkcję:

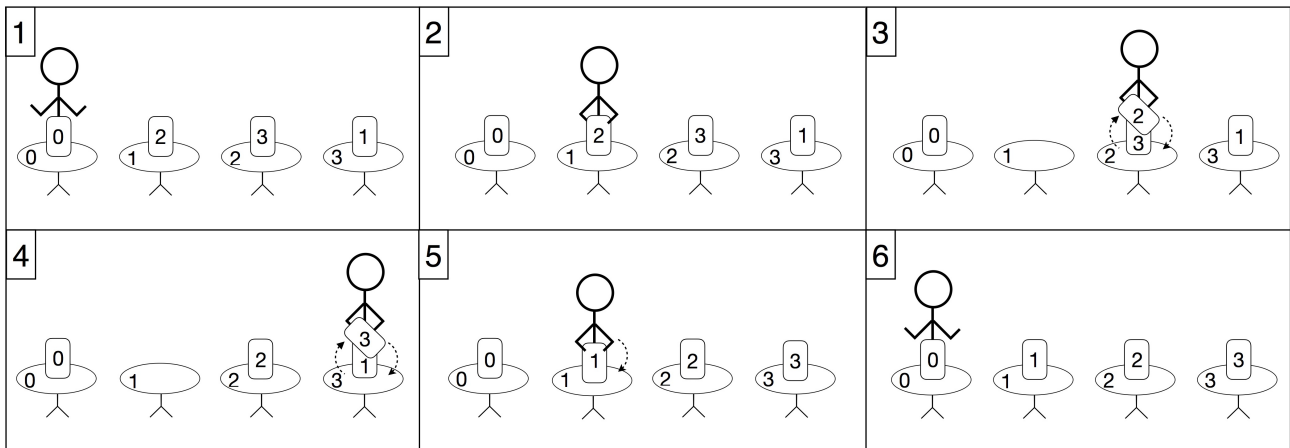
```
int64 minimum_walk(int[] p, int s)
```

- p jest tablicą długości n . Książka, początkowo znajdująca się na stole i , powinna być przeniesiona na stół $p[i]$ (dla każdego $0 \leq i < n$).
- s jest oznaczeniem stołu, gdzie Aryan znajduje się na początku i gdzie powinien powrócić po posortowaniu książek.

- Funkcja powinna zwrócić minimalny łączny dystans (w metrach), który Aryan musi pokonać, aby wykonać zadanie.

Przykład

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



W tym przykładzie, $n = 4$ i Aryan znajduje się przy stole 0 na początku. Sortuje książki w następujący sposób:

- Podchodzi do stołu 1 i podnosi książkę leżącą na nim. Książka ta powinna zostać położona na stole 2.
- Potem przechodzi do stołu 2 i zamienia trzymaną książkę z tą, która leży na stole. Nowa trzymaną książka powinna być odłożona na stole 3.
- Następnie podchodzi do stołu 3 i zamienia trzymaną książkę z tą, która leży na stole. Nowa trzymaną książka powinna być odłożona na stole 1.
- Następnie wraca do stołu 1 i odkłada trzymaną książkę.
- Na końcu wraca do stołu 0.

Zauważ, że książka na stole 0 znajduje się już na właściwym miejscu (stole 0), więc Aryan nie musi jej podnosić. Łączny przebyty dystans w tym rozwiązaniu wynosi 6 metrów. Jest to optymalne rozwiązanie, a zatem funkcja powinna zwrócić 6.

Ograniczenia

- $1 \leq n \leq 1\,000\,000$
- $0 \leq s \leq n - 1$
- W tablicy p znajduje się n parami różnych liczb naturalnych pomiędzy 0 a $n - 1$ włącznie.

Podzadania

1. (12 punktów) $n \leq 4$ i $s = 0$

2. (10 punktów) $n \leq 1000$ i $s = 0$
3. (28 punktów) $s = 0$
4. (20 punktów) $n \leq 1000$
5. (30 punktów) brak dodatkowych ograniczeń

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka odczytuje wejście w następującym formacie:

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

Przykładowa sprawdzaczka wypisuje pojedynczy wiersz zawierający wartość zwróconą przez funkcję `minimum_walk`.