

# Counting Mushrooms (mushrooms)

Expertul în ciuperci Andrew investighează ciupercile native din Singapore.

Ca parte a cercetării sale, Andrew culege  $n$  ciuperci marcate de la 0 la  $n - 1$ . Fiecare ciupercă aparține uneia din cele două specii numite A și B.

Andrew știe că **ciuperca 0 aparține speciei A**, dar, pentru că cele două specii arată la fel, el nu cunoaște speciile ciupercilor de la 1 la  $n - 1$ .

Din fericire, Andrew are o mașină în laborator care îl poate ajuta cu asta. Pentru a folosi această mașină, trebuie plasate două sau mai multe ciuperci în linie în interiorul mașinii și apoi pornită mașina. Apoi, mașina calculează numărul perechilor **adiacente** de ciuperci ce sunt din specii diferite. De exemplu, dacă punem ciuperci din speciile  $[A, B, B, A]$  (în această ordine) în mașină, rezultatul va fi 2.

Totuși, pentru că operarea mașinii este foarte scumpă, mașina poate fi folosit pentru un număr limitat de teste. În plus, numărul total de ciuperci amplasate în mașină cumulat pentru toate testele nu poate depăși 100 000. Folosiți această mașină pentru a-l ajuta pe Andrew să numere ciupercile adunate din specia A.

## Detalii de implementare

Trebuie să implementați următoarea funcție:

```
int count_mushrooms(int n)
```

- $n$ : numărul de ciuperci adunate de Andrew.
- Această funcție este apelată o singură dată, și trebuie să returneze numărul de ciuperci din specia A.

Funcția poate face apeluri către funcția:

```
int use_machine(int[] x)
```

- $x$ : un vector de lungimea între 2 și  $n$  inclusiv, reprezentând etichetele ciupercilor plasate în mașină, în ordine.
- Elementele lui  $x$  trebuie să fie întregi **distincti** de la 0 la  $n - 1$  inclusiv.
- Această funcție poate fi apelată de cel mult 20 000 ori.
- Lungimea totală a lui  $x$  pasată către `use_machine` în toate apelurile nu poate depăși

100 000.

## Exemple

### Exemplul 1

Considerăm un scenariu în care avem 3 ciuperci din speciile  $[A, B, B]$ , în această ordine. Funcția `count_mushrooms` este apelată astfel:

```
count_mushrooms(3)
```

Această funcție poate apela `use_machine([0, 1, 2])`, care (în acest scenariu) returnează 1. Mai apoi, ar putea apela `use_machine([2, 1])`, care returnează 0.

În acest moment, sunt suficiente informații pentru a trage concluzia că există o singură ciupercă din specia A. Astfel, funcția `count_mushrooms` ar trebui să returneze 1.

### Exemplul 2

Considerăm cazul în care avem 4 ciuperci din speciile  $[A, B, A, A]$ , în această ordine. Funcția `count_mushrooms` este apelată astfel:

```
count_mushrooms(4)
```

Funcția poate apela `use_machine([0, 2, 1, 3])`, care returnează 2. Poate apoi apela `use_machine([1, 2])`, care returnează 1.

În acest moment, sunt suficiente informații pentru a trage concluzia că există 3 ciuperci din specia A. Astfel, funcția `count_mushrooms` ar trebui să returneze 3.

## Restricții

- $2 \leq n \leq 20\,000$

## Punctare

În toate testele, dacă apelurile funcției `use_machine` nu sunt conform regulilor de mai sus, sau valoarea returnată de `count_mushrooms` este incorectă, scorul soluției va fi 0. Altfel, fie  $Q$  numărul maxim de apeluri al funcției `use_machine`. Atunci, scorul va fi calculat după următorul tabel:

| Condiție                   | Scor                      |
|----------------------------|---------------------------|
| $20\,000 < Q$              | 0                         |
| $10\,010 < Q \leq 20\,000$ | 10                        |
| $904 < Q \leq 10\,010$     | 25                        |
| $226 < Q \leq 904$         | $\frac{226}{Q} \cdot 100$ |
| $Q \leq 226$               | 100                       |

Pentru unele cazuri comportamentul graderului este adaptiv. Asta înseamnă că pentru aceste teste graderul nu are o secvență fixă a speciilor de ciuperci. În schimb, răspunsul dat de grader poate depinde de apelurile anterioare ale funcției `use_machine`. Totuși, se garantează că răspunsul graderului este astfel încât după fiecare interacțiune există cel puțin o secvență de ciuperci consistentă cu toate răspunsurile date până acum.

## Sample grader

Sample graderul citește un vector de întregi  $s$  reprezentând speciile de ciuperci. Pentru toate valorile  $0 \leq i \leq n - 1$ ,  $s[i] = 0$  înseamnă că specia ciupercii  $i$  este A, iar  $s[i] = 1$  înseamnă că specia ciupercii  $i$  este B. Graderul citește în următorul format:

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

leșirea sample graderului este în următorul format:

- line 1: valoarea returnată de `count_mushrooms`.
- line 2: numărul de apeluri ale `use_machine`.

Sample graderul nu este adaptiv.