



Counting Mushrooms (mushrooms)

Andrew is een paddenstoelenexpert. Hij doet onderzoek naar inheemse paddenstoelen in Singapore.

Voor zijn onderzoek verzamelde Andrew n paddenstoelen genummerd van 0 tot en met $n - 1$. Elke paddenstoel is van één van twee mogelijke types; die worden aangeduid met A of B.

Andrew weet dat **paddenstoel 0 type A is**, maar omdat beide soorten er hetzelfde uitzien, weet hij de types van de paddenstoelen 1 tot en met $n - 1$ niet.

Gelukkig heeft Andrew in zijn laboratorium een machine die hem hierbij kan helpen. Om deze machine te gebruiken dien je twee of meer paddenstoelen in een rij te plaatsen (in een willekeurige volgorde) en de machine aan te zetten. De machine berekent dan het aantal **naast elkaar gelegen** paren paddenstoelen die niet van hetzelfde type zijn. Bijvoorbeeld, als je paddenstoelen van de types $[A, B, B, A]$ (in die volgorde) in de machine plaatst, dan is het resultaat 2.

Omdat het gebruik van de machine heel duur is, mag de machine maar een beperkt aantal keer worden gebruikt. Daarbij mogen er bij die verschillende metingen in totaal maximaal 100 000 paddenstoelen in de machine worden geplaatst. Gebruik deze machine om Andrew te helpen uitzoeken hoeveel paddenstoelen van type A hij heeft verzameld.

Implementatiedetails

Je moet de volgende functie implementeren:

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

- n : het aantal paddenstoelen dat Andrew verzamelde.
- Deze functie wordt precies éénmaal aangeroepen, en moet het aantal paddenstoelen van type A teruggeven.

Deze functie kan de volgende functie aanroepen:

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

- x : een array met een lengte van minstens 2 en ten hoogste n , met daarin de nummers van de paddenstoelen in de volgorde waarin ze in de machine zijn geplaatst.
- De elementen van x moeten **verschillende** integers zijn, elk met een waarde van minstens 0 en ten hoogste $n - 1$.

- Als d de lengte van array x is, dan geeft de functie het aantal verschillende integers j terug waarvoor geldt dat $0 \leq j \leq d - 2$ en dat de paddenstoelen $x[j]$ en $x[j + 1]$ van verschillend type zijn.
- Deze functie mag maximaal 20 000 keer worden aangeroepen.
- De gezamenlijke lengte van alle arrays x die aan de functie `use_machine` worden meegegeven gedurende alle functieaanroepen mag niet groter worden dan 100 000.

Voorbeelden

Voorbeeld 1

Neem het geval dat er 3 paddenstoelen zijn van de typen $[A, B, B]$, in die volgorde. De functie `count_mushrooms` wordt zo aangeroepen:

```
count_mushrooms(3)
```

Deze functie kan `use_machine([0, 1, 2])` aanroepen, wat (in dit scenario) de waarde 1 teruggeeft. Dan kan `use_machine([2, 1])` worden aangeroepen, dat geeft de waarde 0 terug.

Er is dan voldoende informatie beschikbaar om te concluderen dat er 1 paddenstoel is van type A. Daarom moet de functie `count_mushrooms` de waarde 1 teruggeven.

Voorbeeld 2

Neem het geval dat er 4 paddenstoelen zijn van de typen $[A, B, A, A]$, in die volgorde. De functie `count_mushrooms` wordt zo aangeroepen:

```
count_mushrooms(4)
```

Deze functie kan `use_machine([0, 2, 1, 3])` aanroepen, dat geeft (in dit scenario) de waarde 2 terug. Dan kan `use_machine([1, 2])` worden aangeroepen, dat geeft de waarde 1 terug.

Er is dan voldoende informatie beschikbaar om te concluderen dat er 3 paddenstoelen zijn van type A. Daarom moet de functie `count_mushrooms` de waarde 3 teruggeven.

Randvoorwaarden

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

Jurering

Als in een testgeval de aanroepen van de functie `use_machine` niet kloppen met de voorwaarden die hierboven genoemd zijn, of als de teruggegeven waarde van `count_mushrooms` onjuist is, is

de score voor je oplossing 0. Laat anders Q het maximum aantal aanroepen zijn van de functie `use_machine` voor alle testgevallen. Dan wordt de score berekend met de volgende tabel:

Voorwaarde	Score
$20\,000 < Q$	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

In sommige testgevallen is de grader adaptief. Dat wil zeggen dat bij die testgevallen de grader geen van te voren vastgelegde volgorde van paddenstoeltypes heeft; in plaats daarvan hangen de antwoorden van de grader af van eerdere aanroepen van `use_machine`. Het is gegarandeerd dat de grader antwoorden geeft zodat er na iedere aanroep nog steeds minstens één rij aan paddenstoelen bestaat die past bij alle antwoorden die tot op dat moment zijn gegeven.

Voorbeeldgrader

De voorbeeld grader leest een array van integers s in die de paddenstoelentypes aanduiden. Voor alle $0 \leq i \leq n - 1$ betekent $s[i] = 0$ dat het type van paddenstoel i gelijk is aan A, terwijl $s[i] = 1$ betekent dat het type van paddenstoel i gelijk is aan B. De voorbeeldgrader leest invoer in het volgende formaat:

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

De uitvoer van de voorbeeldgrader is in het volgende formaat:

- regel 1: de waarde teruggegeven door `count_mushrooms`.
- regel 2: het aantal aanroepen van `use_machine`.

Let op: de voorbeeldgrader is niet adaptief.