

# Contando Cogumelos (mushrooms)

Andrew, o especialista em cogumelos, está a investigar os cogumelos nativos de Singapura.

Como parte da sua investigação, o Andrew colheu (apanhou)  $n$  cogumelos numerados de 0 a  $n - 1$ . Cada cogumelo é de uma de duas espécies, que são chamadas de A e B.

O Andrew sabe que o **cogumelo 0 pertence à espécie A**, mas como as duas espécies têm o mesmo aspeto, ele não sabe a que espécie pertencem os cogumelos 1 a  $n - 1$ .

Por sorte, o Andrew tem uma máquina que pode ajudar nisto. Para usar a máquina, deve colocar-se dois ou mais cogumelos numa linha (em sequência) na máquina (em qualquer ordem) e ligar a máquina. Então, a máquina calcula o número de pares **adjacentes** de cogumelos que são de diferentes espécies. Por exemplo, se colocares os cogumelos de espécies  $[A, B, B, A]$  (nesta ordem) na máquina, o resultado será 2.

Contudo, como usar a máquina é muito dispendioso, ela apenas pode ser usada um número limitado de vezes. Adicionalmente, o número total de cogumelos colocados na máquina ao longo de todos os seus usos não pode exceder 100 000. Usa a máquina de forma a ajudar o Andrew a contar o número de cogumelos da espécie A que foram colhidos.

## Detalhes de implementação

Deves implementar a seguinte função:

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

- $n$ : número de cogumelos colhidos pelo Andrew.
- Esta função deve ser chamada exatamente uma vez, e deve devolver o número de cogumelos da espécie A.

A função atrás descrita pode fazer chamadas à seguinte função:

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

- $x$ : um array de tamanho entre 2 e  $n$ , inclusive, descrevendo os índices dos cogumelos colocados na máquina, na ordem em que os colocas.
- Os elementos de  $x$  devem ser inteiros **distintos** entre 0 e  $n - 1$ , inclusive.
- Seja  $d$  o tamanho do array  $x$ . Então, a função devolve o número de diferentes índices  $j$  tais que  $0 \leq j \leq d - 2$  e os cogumelos  $x[j]$  e  $x[j + 1]$  são de diferentes espécies.

- Esta função pode ser chamada no máximo 20 000 vezes.
- O tamanho total de  $x$  que é passado à função `use_machine` entre todas as suas chamadas não pode exceder 100 000.

## Exemplos

### Exemplo 1

Considera um cenário onde existem 3 cogumelos das espécies  $[A, B, B]$ , por esta ordem. A função `count_mushrooms` é chamada da seguinte maneira:

```
count_mushrooms(3)
```

Esta função poderia chamar `use_machine([0, 1, 2])`, que por sua vez (neste cenário) iria devolver 1. De seguida poderia chamar `use_machine([2, 1])`, que devolveria 0.

Neste ponto, existe informação suficiente para concluir que existe apenas 1 cogumelo da espécie A. Portanto, a função `count_mushrooms` deveria devolver 1.

### Exemplo 2

Considera um caso onde existe 4 cogumelos das espécies  $[A, B, A, A]$ , por esta ordem. A função `count_mushrooms` é chamada da seguinte maneira:

```
count_mushrooms(4)
```

Esta função poderia chamar `use_machine([0, 2, 1, 3])`, que devolveria 2. De seguida poderia chamar `use_machine([1, 2])`, que devolveria 1.

Neste ponto, existe informação suficiente para concluir que existem 3 cogumelos da espécie A. Portanto, a função `count_mushrooms` deveria devolver 3.

## Restrições

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

## Pontuação

Se em qualquer um dos casos de teste as chamadas à função `use_machine` não respeitarem as regras atrás mencionadas, ou se o valor de retorno de `count_mushrooms` for incorrecto, a pontuação da tua solução será 0. Caso contrário, seja  $Q$  o maior número de chamadas à função `use_machine` entre todos os casos de teste. Então, a pontuação será calculada de acordo com a seguinte tabela:

Condição	Pontuação
$20\,000 < Q$	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

Nalguns casos de teste o comportamento do avaliador é adaptativo. Isto significa que nestes casos de teste o avaliador não tem uma sequência fixa de espécies de cogumelos. Em vez disso, as respostas dadas pelo avaliador podem depender das chamadas anteriores a `use_machine`. Contudo, é garantido que o avaliador responde de tal maneira que, depois de qualquer interação, existe pelo menos uma sequência de espécies de cogumelos que é consistente com as respostas dadas até então.

## Avaliador exemplo

O avaliador exemplo lê um array  $s$  de inteiros representando as espécies de cogumelos. Para todo o  $0 \leq i \leq n - 1$ ,  $s[i] = 0$  significa que a espécie do cogumelo  $i$  é A, ao passo que  $s[i] = 1$  significa que a espécie do cogumelo  $i$  é B. O avaliador exemplo lê o input no seguinte formato:

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

O avaliador exemplo escreve o output com o seguinte formato:

- linha 1: o valor de retorno de `count_mushrooms`.
- linha 2: o número de chamadas a `use_machine`.

Nota que o avaliador exemplo não é adaptativo.