

شمارش قارچ‌ها (mushrooms)

اندرو، متخصص قارچ‌شناسی، در حال بررسی قارچ‌های محلی سنگاپور است.

او به عنوان بخشی از تحقیقات خود n قارچ را جمع‌آوری کرده که 0 تا $n - 1$ شماره‌گذاری شده‌اند. هر قارچ از یکی از دو نوع A یا B است.

اندرو می‌داند که قارچ 0 به نوع A **تعلق دارد**. ولی چون هر دو نوع قارچ از نظر ظاهری مثل هم هستند، او نمی‌داند نوع قارچ‌های 1 تا $n - 1$ چیست.

خوش‌بختانه اندرو در آزمایشگاهش ماشینی دارد که به او در این کار کمک می‌کند. برای استفاده از این ماشین کافی است دو یا تعداد بیش‌تری قارچ را پشت سر هم (به هر ترتیبی) داخل ماشین قرار داد و ماشین را روشن کرد. آن‌گاه ماشین تعداد جفت قارچ‌های **مجاور** که از دو نوع مختلف هستند را محاسبه می‌کند. به عنوان مثال، اگر قارچ‌هایی از انواع $[A, B, B, A]$ را (به این ترتیب) داخل ماشین قرار دهید، نتیجه 2 خواهد بود.

با این حال، چون استفاده از ماشین بسیار پرهزینه است، می‌توان به دفعات محدودی از آن استفاده کرد. به علاوه، مجموع تعداد قارچ‌های قرارداده‌شده داخل ماشین در کل دفعات استفاده از آن نمی‌تواند از 100 000 بیش‌تر شود. با استفاده از این ماشین به اندرو کمک کنید تا تعداد قارچ‌هایی که از نوع A جمع‌آوری کرده را پیدا کند.

جزئیات پیاده‌سازی

شما باید تابع زیر را پیاده‌سازی کنید:

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

- n : تعداد قارچ‌هایی که اندرو جمع‌آوری کرده است.
- این تابع دقیقاً یک بار فراخوانی می‌شود و باید تعداد قارچ‌های از نوع A را برگرداند.

تابع فوق می‌تواند تابع زیر را صدا بزند:

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

- x : یک آرایه با طولی بین 2 تا n که شماره‌ی قارچ‌های قرارگرفته داخل ماشین را به ترتیب نشان می‌دهد.
- عناصر x باید اعداد صحیح **متمایزی** بین 0 تا $n - 1$ باشند.
- این تابع حداکثر 20 000 بار قابل فراخوانی است.
- مجموع طول آرایه‌ی x که به تابع `use_machine` داده می‌شود، در طی تمام فراخوانی‌ها نباید از 100 000 بیش‌تر شود.

مثال‌ها

مثال ۱

فرض کنید 3 قارچ با انواع $[A, B, B]$ به این ترتیب داریم. تابع `count_mushrooms` به صورت زیر فراخوانی می‌شود:

```
count_mushrooms(3)
```

تابع می‌تواند `use_machine([0, 1, 2])` را صدا بزند، که در این مثال عدد 1 را برمی‌گرداند. سپس می‌تواند `use_machine([2, 1])` را صدا بزند، که عدد 0 را برمی‌گرداند.

در این لحظه، اطلاعات کافی وجود دارد که نتیجه بگیریم فقط یک قارچ از نوع A وجود دارد. در نتیجه تابع `count_mushrooms` باید مقدار 1 را برگرداند.

مثال ۲

حالتی را در نظر بگیرید که 4 قارچ با انواع $[A, B, A, A]$ به این ترتیب داریم. تابع `count_mushrooms` به صورت زیر فراخوانی می‌شود:

```
count_mushrooms(4)
```

تابع می‌تواند `use_machine([0, 2, 1, 3])` را صدا بزند، که در این مثال عدد 2 را برمی‌گرداند. سپس می‌تواند `use_machine([1, 2])` را صدا بزند، که عدد 1 را برمی‌گرداند.

در این لحظه، اطلاعات کافی وجود دارد که نتیجه بگیریم 3 قارچ از نوع A وجود دارد. در نتیجه تابع `count_mushrooms` باید عدد 3 را برگرداند.

محدودیت‌ها

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

امتیازدهی

اگر در هر یک از موارد آزمون، فراخوانی تابع `use_machine` از قوانینی که در بالا توضیح داده شد تبعیت نکند، یا مقدار خروجی `count_mushrooms` نادرست باشد، امتیاز دریافتی 0 خواهد بود. در غیر این صورت، فرض کنید Q بیش‌ترین تعداد فراخوانی `use_machine` در بین تمام موارد آزمون باشد. آنگاه امتیاز به صورت زیر محاسبه می‌شود:

امتیاز	شرط
0	$20\ 000 < Q$
10	$10\ 010 < Q \leq 20\ 000$
25	$904 < Q \leq 10\ 010$
$\frac{226}{Q} \cdot 100$	$226 < Q \leq 904$
100	$Q \leq 226$

در برخی موارد آزمون، رفتار ارزیاب به صورت «تطبیق‌دهنده» خواهد بود. این بدان معنی است که در این موارد آزمون، ترتیب انواع قارچ از ابتدا ثابت نیست. در عوض، پاسخی که توسط ارزیاب داده می‌شود ممکن است به فراخوانی‌های قبلی `use_machine` بستگی داشته باشد. با این حال تضمین می‌شود که پاسخ‌های ارزیاب به نحوی است که پس از هر فراخوانی، حداقل یک دنباله از انواع قارچ‌ها وجود دارد که با پاسخ‌های داده‌شده تا کنون سازگاری دارد.

ارزیاب نمونه

ارزیاب نمونه یک آرایه‌ی s از اعداد صحیح را می‌خواند که انواع قارچ‌ها را مشخص می‌کند. به ازای هر مقدار $0 \leq i \leq n-1$ ، مقدار $s[i] = 0$ بدان معنی است که قارچ i از نوع A است، و $s[i] = 1$ بدان معنی است که قارچ i از نوع B است. ارزیاب نمونه ورودی را در قالب زیر می‌خواند:

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

خروجی ارزیاب نمونه در قالب زیر است:

- خط 1: مقدار خروجی `count_mushrooms`.
- خط 2: تعداد فراخوانی‌های `use_machine`.

دقت کنید که ارزیاب نمونه تطبیق‌دهنده نیست.