



Göbələklərin Sayılması (mushrooms)

Göbələklər mütəxəssisi Andrew Sinqapurdan gətirilən göbələkləri araşdırır.

Tədqiqatının bir hissəsi kimi, Andrew 0-dan $n - 1$ -dək nömrələnmiş n sayda göbələk topladı. Hər bir göbələk A və B adlanan iki növdən birinə aiddir.

Andrew onu bilir ki 0-cı **göbələk A növünə aiddir**, lakin iki növ eyni göründüyünə görə, 1-dan $n - 1$ -dək olan göbələklərin növünü bilmir.

Xoşbəxtlikdən, Andrewin laboratoriyasında buna kömək edə biləcək bir maşın var. Bu maşından istifadə etmək üçün, biri iki və ya daha çox göbələyi maşına sıra ilə yerləşdirir (istənilən ardıcılıqda) və sonra maşını işə salır. Sonra işə, maşın **ardıcıl** yerləşən və fərqli növlərə aid olan göbələk cütlərinin sayını hesablayır. Misal üçün, əgər siz $[A, B, B, A]$ (bu ardıcılıqda) növlərindəki göbələkləri maşına yerləşdirsəniz, cavab 2 olacaq.

Lakin, maşını işlətmək baha başa gəldiyinə görə, maşın məhdud sayda istifadə edilə bilər. Əlavə olaraq, bütün istifadələrdə maşına yerləşdirilən toplam göbələklərin sayı 100 000-i aşma bilməz. Andrewa bu maşından istifadə edərək A növündə toplanılan göbələklərin sayını tapmaqda kömək edin.

İmplementasiya detalları

Aşağıdakı proseduru implement etməlisiniz:

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

- n : Andrew toplayan göbələklərin sayı.
- Bu prosedur yalnız bir dəfə çağırılır, və A növünə aid olan göbələklərin sayını qaytarmalıdır.

Yuxarıdakı prosedur aşağıdakı proseduru çağırma bilər:

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

- x : Maşına ardıcılıqla yerləşdirilən göbələklərin etiketlərin təsvir edən, uzunluğu 2 və n (n daxil olmaqla) arasında olan massiv.
- x -in elementləri 0-dan $n - 1$ -dək ($n - 1$ daxil olmaqla) **fərqli** tam ədədlərdir.
- Bu prosedur ən çoxu 20 000 dəfə çağırılma bilər.
- Bütün çağırışlar zamanı use_machine proseduruna ötürülən bütün x -lərin toplam uzunluqları 100 000-i aşma bilməz.

Nümunələr

Nümunə 1

3 dənə $[A, B, B]$ növündə və ardıcılığında olan göbələk ssenarini nəzərdən keçirin. `count_mushrooms` proseduru aşağıdakı kimi çağırılıb:

```
count_mushrooms(3)
```

Bu prosedur `use_machine([0, 1, 2])` çağıra bilər, hansı ki (bu ssenari üçün) 1 qaytarır. Bundan sonra o `use_machine([2, 1])` çağıra bilər, hansı ki 0 qaytarır.

Bu nöqtədə, A növündən 1 dənə göbələk olmağı qənaətinə gəlmək üçün kifayət qədər məlumat var . Beləliklə, `count_mushrooms` proseduru 1 qaytarmalıdır.

Nümunə 2

4 dənə $[A, B, A, A]$ növündə və ardıcılığında olan göbələk halını nəzərdən keçirin . `count_mushrooms` proseduru aşağıdakı kimi çağırılıb:

```
count_mushrooms(4)
```

Bu prosedur `use_machine([0, 2, 1, 3])` çağıra bilər, hansı ki 2 qaytarır. Bundan sonra o `use_machine([1, 2])` çağıra bilər, hansı ki 1 qaytarır.

Bu nöqtədə, A növündən 3 dənə göbələk olmağı qənaətinə gəlmək üçün kifayət qədər məlumat var . Beləliklə, `count_mushrooms` proseduru 3 qaytarmalıdır.

Məhdudiyyətlər

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

Qiymətləndirmə

Əgər hər hansı bir test halında, `use_machine`-ə edilən çağırışlar yuxarıda göstərilən qaydalara uyğun gəlmərsə, və ya `count_mushrooms`-un qayıdış dəyəri səhvdirsə, həlliniz 0 xal qazanacaq. Əks halda, fərz edək ki Q bütün test hallar arasında `use_machine`-ə edilən çağırışların maksimum sayıdır. Onda, xal aşağıdakı cədvələ əsasən hesablanacaqdır:

Şərt	Xal
$20\,000 < Q$	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

Bəzi test hallarda, greyderin davranışı adaptivdir. Bu o deməkdir ki, bu test hallarda graderin müəyyən bir göbələk növü ardıcılığı yoxdur. Bunun əvəzinə, grader tərəfdən verilən cavab `use_machine`-ə əvvəlcədən edilən çağırışlardan asılı ola bilər. Buna baxmayaraq, qarantiya var ki, grader elə cavab verəcək ki, hər qarşılıqlı təsirdən sonra indiyə qədər verilən bütün cavablara uyğun ən azı bir göbələk növü ardıcılığı olacaq.

Nümunə grader

Nümunə grader göbələklərin növlərini verən tam ədədlər s massivini oxuyur. Bütün $0 \leq i \leq n - 1$, $s[i] = 0$ o deməkdir ki i göbələyin növü A-dır, $s[i] = 1$ isə o deməkdir ki i göbələyin növü B-dir. Nümunə grader giriş verilənlərini aşağıdakı formatda oxuyur:

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

Nümunə graderin çıxışı aşağıdakı formatdadır:

- sətir 1: `count_mushrooms`-un qayıdış dəyəri.
- sətir 2: `use_machine`-nin çağırış sayı .

Unutmayın ki nümunə grader adaptiv deyil.