

Permutasiya

Fironlar öz kosmik gəmilərini sürətləndirmək üçün planetlərin nisbi hərəkətindən və cazibə qüvvəsindən istifadə edirlər. Fərz edək ki, kosmik gəmi orbital sürətləri $p[0], p[1], \dots, p[n-1]$ olan n planetin yanından keçəcək. Hər bir planet üçün Fironların alimləri bu planetdən istifadə edərək kosmik gəmini sürətləndirib-sürətləndirməməyə qərar verə bilirlər. Enerjiyə qənaət etmək üçün orbital sürəti $p[i]$ olan bir planet tərəfindən sürətləndirildikdən sonra kosmik gəmi orbital sürəti $p[j] < p[i]$ olan heç bir planetdən istifadə edərək sürətləndirilə bilməz. Başqa sözlə, seçilmiş planetlər $p[0], p[1], \dots, p[n-1]$ -in **artan alt ardıcılığı**nı təşkil edir. p -nin alt ardıcılığı p -dən sıfır və ya daha çox sayda elementini silinməklə əldə edilən ardıcılıqdır. Məsələn, $[0]$, $[], [0, 2]$ və $[0, 1, 2]$ $[0, 1, 2]$ -nin alt ardıcılıqlarıdır, lakin $[2, 1]$ deyil.

Alimlər müəyyən ediblər ki, kosmik gəmini sürətləndirmək üçün cəmi k müxtəlif üsulla bir sıra planetləri seçmək olar, lakin onlar bütün orbital sürət qeydlərini itiriblər. (hətta n -in dəyərini də). Bununla belə, onlar xatırlayırlar ki, $(p[0], p[1], \dots, p[n-1])$ $(0, 1, \dots, n-1)$ -in permutasiyasıdır. Permutasiya 0-dan $n-1$ -ə qədər hər bir tam ədədi bir dəfə ehtiva edən ardıcılıqdır. Sizin tapşırığınız kifayət qədər kiçik uzunluqda mümkün bir permutasiya $(p[0], p[1], \dots, p[n-1])$ tapmaqdır.

Siz q sayda müxtəlif kosmik gəmi üçün problemi həll etməlisiniz. Hər i kosmik gəmisi üçün sizə k_i tam ədədi verilir, bu, kosmik gəmini sürətləndirmək üçün bir sıra planetlərin seçilə biləcəyi müxtəlif üsulların sayını bildirir. Sizin tapşırığınız kifayət qədər kiçik n_i uzunluğuna malik elə orbital sürətlər ardıcılığını tapmaqdır ki, artan orbital sürətlərə malik planetlərin alt ardıcılığını seçmək üçün tam olaraq k_i müxtəlif üsul olsun.

İmplementasiya Detalları

Aşağıdakı proseduru icra etməlisiniz:

```
int[] construct_permutation(int64 k)
```

- k : artan alt ardıcılıqların arzu olunan sayıdır.
- Bu prosedur hər bir elementi 0 və $n-1$ arasında olan n elementli massiv qaytarmalıdır.
- Qaytarılan massiv tam olaraq k artan alt ardıcılığa malik permutasiya olmalıdır.
- Bu prosedur cəmi q dəfə çağırılır. Bu çağırışların hər birinə ayrıca ssenari kimi yanaşmaq lazımdır.

Məhdudiyyətlər

- $1 \leq q \leq 100$

- $2 \leq k_i \leq 10^{18}$ (for all $0 \leq i \leq q - 1$)

Alt Tapşırıqlar

1. (10 bal) $2 \leq k_i \leq 90$ (bütün $0 \leq i \leq q - 1$ üçün). İstifadə etdiyiniz bütün permutasiyaların uzunluğu ən çoxu 90-dırsa və düzgündürsə, siz 10 bal, əks halda 0 bal alırsınız.
2. (90 bal) Əlavə məhdudiyyətlər yoxdur. Bu alt tapşırıq üçün gəlin m bütün ssenarilərdə istifadə etdiyiniz maksimum permutasiya uzunluğu olsun. Bu halda, balınız aşağıdakı cədvələ uyğun olaraq hesablanır:

Şərt	Bal
$m \leq 90$	90
$90 < m \leq 120$	$90 - \frac{(m-90)}{3}$
$120 < m \leq 5000$	$80 - \frac{(m-120)}{65}$
$m > 5000$	0

Nümunələr

Nümunə 1

Aşağıdakı prosedura nəzər yetirək:

```
construct_permutation(3)
```

Bu prosedur tam olaraq 3 artan alt ardıcılığa malik bir permutasiya qaytarmalıdır. Mümkün cavablardan biri $[1, 0]$ -dir. Onun $[]$ (boş ardıcılıq), $[0]$ və $[1]$ kimi artan alt ardıcılıqları var.

Nümunə 2

Aşağıdakı prosedura nəzər yetirək:

```
construct_permutation(8)
```

Bu prosedur tam olaraq 8 artan alt ardıcılığa malik bir permutasiya qaytarmalıdır. Mümkün cavablardan biri $[0, 1, 2]$ -dir.

Nümunə qreyder

Nümunə qreyder giriş verilənlərini aşağıdakı formatda oxuyur:

- sətir 1: q
- sətir $2 + i$ ($0 \leq i \leq q - 1$): k_i

Nümunə qreyder hər k_i üçün `construct_permutation` prosedurunun qaytardığı dəyəri və ya əgər baş vermişsə, xəta mesajını ehtiva edən yeganə sətir çap edir.