

Hack! (hack)

Codeforces-də yarışın başlamasından bir saat sonra siz otağınızda ki başqa bir iştirakçının bir məsələni `unordered_set` istifadə edərək həll etdiyini gördünüz. İndi isə hack vaxtıdır!

Siz bilirsiniz ki, `unordered_set` bir hash cədvəlindən istifadə edir və bu cədvəl n ədəd bəket (qab) ehtiva edir, bunlar 0-dan $n - 1$ -ə qədər nömrələnib. Təəssüf ki, n dəyərini bilmirsiniz və onu tapmaq istəyirsiniz.

x tam ədədi hash cədvəlinə əlavə edildikdə, o, $(x \bmod n)$ -ci bəketə yerləşdirilir. Əgər bu bəketdə əlavə etməzdən əvvəl b element varsa, bu, b ədəd hash toqquşması (collision) ilə nəticələnəcək.

Əgər interaktora k fərqli tam ədəd $x[0], x[1], \dots, x[k - 1]$ ötürsəniz, o zaman bu ədədlərdən ibarət bir `unordered_set` qurularkən ümumilikdə neçə toqquşma baş verdiyini öyrənə bilərsiniz. Lakin bir sorğuda interaktora k ədəd ötürmək k qədər xərc yaradır.

Məsələn, əgər $n = 5$ olarsa, və interaktora $x = [2, 15, 7, 27, 8, 30]$ ötürsəniz, ümumilikdə 4 toqquşma baş verəcək:

Əməliyyat	Yeni toqquşmalar	Bəketlər
<i>başlanğıcda</i>	—	$[], [], [], [], []$
$x[0] = 2$ əlavə et	0	$[], [], [2], [], []$
$x[1] = 15$ əlavə et	0	$[15], [], [2], [], []$
$x[2] = 7$ əlavə et	1	$[15], [], [2, 7], [], []$
$x[3] = 27$ əlavə et	2	$[15], [], [2, 7, 27], [], []$
$x[4] = 8$ əlavə et	0	$[15], [], [2, 7, 27], [8], []$
$x[5] = 30$ əlavə et	1	$[15, 30], [], [2, 7, 27], [8], []$

Qeyd: interaktor hash cədvəlini verilmiş elementləri sırasıyla boş bir `unordered_set`-ə daxil edərək yaradır və hər sorğu üçün yeni bir boş `unordered_set` istifadə olunur. Başqa sözlə, bütün sorğular müstəqildir.

Sizin tapşırığınız – ümumi xərc 1 000 000-u keçmədən bəketlərin sayını, yəni n -i tapmaqdır.

İmplementasiya detalları

Aşağıdakı proseduru icra etməlisiniz:

```
int hack()
```

- Bu prosedur bir tam ədəd qaytarmalıdır – gizli olan n dəyərini.
- Hər test halı üçün, qiymətləndirici bu proseduru bir neçə dəfə çağırır. Hər çağırışa tamamilə yeni bir ssenari kimi yanaşmaq lazımdır.

Bu prosedur daxilində aşağıdakı proseduru çağırır bilərsiniz:

```
long long collisions(std::vector<long long> x)
```

- x : fərqli ədədlərdən ibarət massiv, burada $1 \leq x[i] \leq 10^{18}$.
- Bu prosedur, x -in elementlərini `unordered_set`-ə daxil etdikdə ümumilikdə neçə toqquşma baş verdiyini qaytarır.
- Bu prosedur bir neçə dəfə çağırılır. `hack()` proseduruna bir çağırış daxilində $\$x\$$ massivlərinin uzunluqlarının cəmi 1 000 000-u keçməməlidir.

Qeyd: `hack()` proseduru bir neçə dəfə çağırılacağından, iştirakçılar əvvəlki çağırışlardan qalan məlumatların cari çağırışa təsirini, xüsusilə də qlobal dəyişənlərdə saxlanılan halları nəzərə almalıdırlar.

1 000 000 xərc limiti hər bir `hack()` çağırışına aiddir. Ümumilikdə, əgər t ədəd `hack()` çağırışı varsa, sizin ümumi xərciniz $t \times 1\,000\,000$ -u keçməməlidir, eyni zamanda hər bir `hack()` çağırışı üçün sərf edilən xərc 1 000 000-u keçməməlidir.

İnteraktor adaptiv deyil, yəni n dəyərləri interaksiya başlamazdan əvvəl təyin olunur.

Nümunə

Tutaq ki, 2 multitest var. Qiymətləndirici aşağıdakı çağırışı edəcək:

```
hack()
```

Deyək ki, bu prosedur daxilində aşağıdakı çağırışları etdiniz:

Çağırış	Qaytarılan dəyər
<code>collisions([2, 15, 7, 27, 8, 30])</code>	4
<code>collisions([1, 2, 3])</code>	0
<code>collisions([10, 20, 30, 40, 50])</code>	10

Bundan sonra əgər siz n -in dəyərinin 5 olduğunu tapdınsa, onda `hack()` proseduru 5 qaytarmalıdır.

Sonra qiymətləndirici başqa bir çağırış edəcək:

```
hack()
```

Deyək ki, bu dəfə aşağıdakı çağırışları etdiniz:

Çağırış	Qaytarılan dəyər
<code>collisions([1, 3])</code>	1
<code>collisions([2, 4])</code>	1

Bu sorğuları ödəyən yeganə n dəyəri 2-dir. Beləliklə, `hack()` proseduru 2 qaytarmalıdır.

Məhdudiyyətlər

- $1 \leq t \leq 10$, burada t multitestlərin sayıdır.
- $2 \leq n \leq 10^9$
- Hər bir `collisions()` çağırışında $1 \leq x[i] \leq 10^{18}$.

Alt Tapşırıqlar

- (8 bal) $n \leq 500\,000$
- (17 bal) $n \leq 1\,000\,000$
- (75 bal) Əlavə məhdudiyyət yoxdur.

Son alt tapşırıqda qismən bal almaq mümkündür. Qoy q alt tapşırıq daxilindəki bütün `hack()` çağırışları arasında maksimum ümumi xərci göstərsin. Bu alt tapşırıq üçün sizin balınız aşağıdakı cədvələ əsasən hesablanır:

Şərt	Bal
$1\,000\,000 < q$	0
$110\,000 < q \leq 1\,000\,000$	$75 \cdot \log_{50} \left(\frac{10^6}{x-90000} \right)$
$q \leq 110\,000$	75

Əgər hər hansı test halında `collisions()` proseduruna çağırışlar implementasiya detallarında göstərilən məhdudiyyətlərə uyğun deyilsə və ya `hack()` düzgün dəyər qaytarmırsa, həmin alt tapşırıq üçün sizin balınız 0 olacaq.

Nümunə Qiymətləndirici

Nümunə qiymətləndirici giriş verilənlərini aşağıdakı formatda oxuyur:

- 1-ci sətir: t

Sonra t sətir gəlir, hər biri n dəyərini ehtiva edir:

- 1-ci sətir: n

Hər bir test halı üçün qoy m `hack()` -dən qaytarılan dəyər, c isə bütün sorğuların ümumi xərci olsun. Nümunə qiymətləndirici cavabınızı aşağıdakı formatda çap edir:

- 1-ci sətir: m c