

Hack! (hack)

Codeforces sınavında odanızdaki bir yarışmacının çözümünde `unordered_set` kullandığını gördünüz ve çözümü hacklemeye karar verdiniz.

Biliyorsunuz ki `unordered set` n kova (bucket) içeren hash table kullanır, kovalar 0 'dan $n - 1$ 'e numaralandırılır. Ne yazık ki n değerini bilmiyorsun ve bulmak istiyorsun.

x sayısını hash table'a eklediğinizde, sayı $(x \bmod n)$ -inci kovaya eklenir . Eğer kovada, ekleme yapmadan önce b eleman varsa b hash çakışması (collisions) gerçekleşir.

İnteraktöre k farklı tamsayı $x[0], x[1], \dots, x[k - 1]$ verdiğinizde, interaktör bu sayıları sırasıyla kümeye eklerken oluşan toplam çakışma sayısını döner. İnteraktöre bir sorguda k tamsayı vermek k maliyete sebep olur.

Örneğin, eğer $n = 5$ ise, interaktöre $x = [2, 15, 7, 27, 8, 30]$ tamsayılarını vermek toplamda 4 çakışmaya sebebiyet verir:

Operasyon	Yeni Çakışmalar	Kovalar
<i>başlangıçta</i>	—	$[], [], [], [], []$
$x[0] = 2$ ekle	0	$[], [], [2], [], []$
$x[1] = 15$ ekle	0	$[15], [], [2], [], []$
$x[2] = 7$ ekle	1	$[15], [], [2, 7], [], []$
$x[3] = 27$ ekle	2	$[15], [], [2, 7, 27], [], []$
$x[4] = 8$ ekle	0	$[15], [], [2, 7, 27], [8], []$
$x[5] = 30$ ekle	1	$[15, 30], [], [2, 7, 27], [8], []$

Unutmayın ki, interaktör elemanları sırasıyla başlangıçta boş olan sete teker teker ekleyerek hash table'ı oluşturur. Ayrıca her yeni sorguda set ve hash table tekrar boş şekilde başlar. Özetle her sorgu birbirinden bağımsızdır.

n sayısını en fazla 1 000 000 maliyetle tespit ediniz.

Kodlama Detayları

Aşağıdaki prosedürü kodlamalısınız

```
int hack()
```

- Bu prosedür bir tamsayı döndürmeli – n değeri.
- Her test için, grader bu fonksiyonu birden çok kere çağırabilir. Her çağrı yeni bir senaryo gibi işlenmelidir.

Bu prosedürde aşağıdaki prosedürü çağırabilirsiniz:

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

- x : farklı sayılar içeren bir dizi, her i için $1 \leq x[i] \leq 10^{18}$.
- Bu fonksiyon x dizisindeki elemanları unordered set'e eklerken oluşan toplam çakışma sayısını döndürür.
- Bu prosedür birden çok kere çağırılabilir. Bir `hack()` fonksiyonu için, fonksiyona gönderilen x dizilerinin uzunluğu en fazla 1 000 000 olabilir.

Not: `hack()` fonksiyonu birden çok kere çağırılabilir için, yarışmacılar önceki çağrıdan kalan, başta global veriler olmak üzere verilere dikkat etmelidir.

1 000 000 maliyet limiti her test için ayrı uygulanır. Özetle, eğer t `hack()` çağrısı yapıldıysa, toplam maliyet $t \times 1\,000\,000$ değerini geçmemelidir, ayrıca her bireysel `hack()` çağrısının maliyeti en fazla 1 000 000 olmalıdır.

Örnek

Varsayalım ki, en fazla 2 çoklute test yapısı olsun. grader şu çağrıyı yapacaktır:

```
hack()
```

Diyelim ki fonksiyonda şu çağrıları yapıyoruz:

Çağrı	Dönen Değer
<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, eğer n değerini 5 bulursanız, `hack()` fonksiyonu 5 dönmelidir.

Daha sonra grader başka bir çağrı yapacaktır:

```
hack()
```

Diyelim ki şu çağrılar yaptınız:

Çağrı	Dönen Değer
<code>collisions([1, 3])</code>	1
<code>collisions([2, 4])</code>	1

n değerinin alabileceği tek değer 2 olabilir. Bu sebeple, `hack()` 2 değerini dönmelidir.

Kısıtlar

- $1 \leq t \leq 10$, t çoklu testlerin sayısını verir.
- $2 \leq n \leq 10^9$
- $1 \leq x[i] \leq 10^{18}$ her `collisions()` çağrısı için.

Alt Görevler

1. (8 puan) $n \leq 500\,000$
2. (17 puan) $n \leq 1\,000\,000$
3. (75 puan) Hiçbir ek kısıt yoktur.

Son alt görevde parçalı puan almak mümkündür. Diyelim ki q , alt görevdeki bütün `hack()` fonksiyonları için kullandığınız maksimum maliyet olsun. Bu durumda skorunuz tabloda verilen şekilde hesaplanacaktır:

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

Eğer herhangi bir testte `collisions()` fonksiyonuna gönderdiğiniz değerler Kodlama Detayları kısmında belirtilen ifadelerle uyuşmazsa veya `hack()` fonksiyonunda döndürdüğünüz değer hatalıysa, ilgili alt görevden alacağınız puan 0 olur.

Örnek Grader

Örnek grader girdiyi şu formatta okur:

- line 1: t

Sonra, t satır takip eder, her satırda n değeri bulunur:

- line 1: n

Her test için, $m_{\text{hack}}()$ fonksiyonunun döneceği değer olsun, ve c bütün sorguların kullandığı toplam maaliyet olsun. Örnek grader cevabınızı şu formatta yazdırır:

- line 1: $m\ c$