



## Nadir Böcekler

PakBlangkon'un evinin etrafında dolaşan 0 ile  $N - 1$  arasında indekslenmiş  $N$  tane böcek vardır. Her böceğin bir **türü** vardır ve böcek türü 0 ile  $10^9$  arasında bir tam sayıdır. Birden fazla böcek aynı türe sahip olabilir.

Böceklerin türlerine göre gruplandırıldığını varsayalım. **En sık görülen** böcek türünün eleman sayısı, en fazla böcek sayısına sahip olan gruptaki böcek sayısı olarak tanımlanır. Benzer şekilde, **en nadir** böcek türünün eleman sayısı, en az böcek sayısına sahip olan gruptaki böcek sayısıdır.

Örneğin, türleri  $[5, 7, 9, 11, 11, 5, 0, 11, 9, 100, 9]$  olan 11 tane böcek olduğunu varsayalım. Bu durumda, **en sık** böcek türünün eleman sayısı 3'dür. En fazla sayıda böcek içeren gruplar, her biri 3 tane böcek içeren ve türleri 9 ve 11 gruplardır. **En nadir** böcek türünün eleman sayısı 1'dir. En az sayıda böcek içeren gruplar, her biri 1 tane böcek içeren ve türleri 7, 0 ve 100 olan gruplardır.

PakBlangkon herhangi bir böceğin türünü bilmemektedir. PakBlangkon'ın böceklerin türleri hakkında bilgi verebilecek tek düğmeli bir makinesi vardır. Başlangıçta, makine boştur. Makineyi kullanmak için üç tip işlem gerçekleştirilebilir:

1. Bir böceği makinenin içine taşı.
2. Bir böceği makinenin dışına çıkar.
3. Makinedeki düğmeye bas.

Her bir işlem türü en fazla 40 000 kez gerçekleştirilebilir.

Düğmeye her basıldığında, makine yalnızca makinenin içindeki böcekleri dikkate alarak **en sık** böcek türünün eleman sayısını bildirir.

Göreviniz, makineyi kullanarak PakBlangkon'un evindeki tüm  $N$  tane böcek arasında **en nadir** böcek türünün eleman sayısını belirlemektir. Ek olarak, bazı alt görevlerde puanınız, belirli bir tipte gerçekleştirilen maksimum işlem sayısına bağlıdır (ayrıntılar için Alt Görevler bölümüne bakın).

## Programlama Detayları

Aşağıdaki fonksiyonu kodlamalısınız:

```
int min_cardinality(int N)
```

- $N$ : böcek sayısı.

- Bu fonksiyon, PakBlangkon'un evindeki tüm  $N$  tane böcek arasında **en nadir** böcek türünün eleman sayısını return etmelidir.
- Bu fonksiyon tam olarak bir kez çağrılır.

Yukarıdaki fonksiyon aşağıdaki fonksiyonları çağırabilir:

```
void move_inside(int i)
```

- $i$ : Makine içinde hareket ettirilecek böceğin indeksi.  $i$  değeri 0 ile  $N - 1$  arasında olmalıdır (sınırlar dahil).
- Bu böcek zaten makinenin içindeyse, çağrının makinedeki böcek kümesi üzerinde hiçbir etkisi yoktur. Ancak yine de ayrı bir çağrı olarak sayılır.
- Bu fonksiyon en fazla 40 000 defa çağrılabilir.

```
void move_outside(int i)
```

- $i$ : makinenin dışına taşınacak böceğin indeksi.  $i$  değeri 0 ile  $N - 1$  arasında olmalıdır (sınırlar dahil).
- Bu böcek zaten makinenin dışındaysa, çağrının makinedeki böcek kümesi üzerinde hiçbir etkisi yoktur. Ancak yine de ayrı bir çağrı olarak sayılır.
- Bu fonksiyon en fazla 40 000 defa çağrılabilir.

```
int press_button()
```

- Bu fonksiyon, yalnızca makinenin içindeki böcekleri dikkate alarak **en sık** böcek türünün eleman sayısını return eder.
- Bu fonksiyon en fazla 40 000 defa çağrılabilir.
- Değerlendirici **adaptif değildir**. Yani, tüm  $N$  böceğin türü, `min_cardinality` çağrılmadan önce sabitlenir.

## Örnek

Sırasıyla [5,8,9,5,9,9] türünden oluşan 6 tane böceğin olduğu bir senaryo düşünün. `min_cardinality` fonksiyonu şu şekilde çağrılır:

```
min_cardinality(6)
```

Fonksiyon `move_inside`, `move_outside`, ve `press_button` 'u aşağıdaki gibi çağırabilir.

Çağırım	Return değeri	Makinedeki böcek	Makinedeki böcek türleri
		{}	[]
move_inside(0)		{0}	[5]
press_button()	1	{0}	[5]
move_inside(1)		{0, 1}	[5, 8]
press_button()	1	{0, 1}	[5, 8]
move_inside(3)		{0, 1, 3}	[5, 8, 5]
press_button()	2	{0, 1, 3}	[5, 8, 5]
move_inside(2)		{0, 1, 2, 3}	[5, 8, 9, 5]
move_inside(4)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
move_inside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
press_button()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
move_inside(5)		{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
press_button()	3	{0, 1, 2, 3, 4, 5}	[5, 8, 9, 5, 9, 9]
move_outside(5)		{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]
press_button()	2	{0, 1, 2, 3, 4}	[5, 8, 9, 5, 9]

Bu noktada, en nadir görülen böcek türünün eleman sayısının 1 olduğu sonucuna varmak için yeterli bilgi bulunmaktadır. Bu nedenle, `min_cardinality` fonksiyonu 1 return etmelidir.

Bu örnekte, `move_inside` 7 kez, `move_outside` 1 kez ve `press_button` 6 kez çağrılır.

## Kısıtlar

- $2 \leq N \leq 2000$

## Alt görevler

1. (10 puan)  $N \leq 200$
2. (15 puan)  $N \leq 1000$
3. (75 puan) Ek kısıt yoktur.

Test senaryolarının herhangi birinde, `move_inside`, `move_outside`, veya `press_button` fonksiyonlarına yapılan çağrılar, Programlama Detaylarında açıklanan kısıtlamalara uymuyorsa veya `min_cardinality` 'nin return değeri yanlışsa, bu alt görev için çözümünüz 0 puan olacaktır.

$q$ , şu üç değerin **maksimumu** olsun: `move_inside` 'a yapılan çağrılarının sayısı, `move_outside` 'a yapılan çağrılarının sayısı ve `press_button` 'a yapılan çağrılarının sayısı.

Alt görev 3'te kısmi puan alabilirsiniz. Bu alt görevdeki tüm test senaryolarında  $\frac{q}{N}$ 'in maksimum değeri  $m$  olsun. Bu alt görev için puanınız aşağıdaki tabloya göre hesaplanır:

Koşul	Puan
$20 < m$	0 (CMS 'de "Output isn't correct" (çıkıktı yanlış) olarak belirtilir)
$6 < m \leq 20$	$\frac{225}{m-2}$
$3 < m \leq 6$	$81 - \frac{2}{3}m^2$
$m \leq 3$	75

## Örnek Değerlendirici

$T$ ,  $N$  tane tam sayı içeren bir tamsayı dizisi olsun. Burada,  $T[i]$ 'in  $i$  böceğinin türüdür.

Örnek değerlendirici, girdiyi aşağıdaki formatta okur:

- satır 1:  $N$
- satır 2:  $T[0] T[1] \dots T[N - 1]$

Örnek değerlendirici bir protokol ihlali tespit ederse, örnek değerlendiricinin çıktısı `Protocol Violation: <MSG>` olur, burada `<MSG>` aşağıdakilerden biridir:

- `invalid parameter`: `move_inside` veya `move_outside` 'a yapılan çağrılarının birinde,  $i$ 'nin değeri 0 ve  $N - 1$  arasında değildir (sınırlar dahil).
- `too many calls`: `move_inside`, `move_outside`, veya `press_button` 'a yapılan çağrılarının **herhangi biri** 40 000 'i aşar.

Aksi takdirde, örnek değerlendiricinin çıktısı aşağıdaki formatta olur:

- satır 1: `min_cardinality` 'nin return değeri
- satır 2:  $q$