

# Zindan Oyunu

Robert yeni bir bilgisayar oyunu tasarlıyor. Bu oyunda bir kahraman,  $n$  düşman ve  $n + 1$  zindan bulunmaktadır. Düşmanlar  $0$  ile  $n - 1$  arasında ve zindanlar  $0$  ile  $n$  arasında numaralandırılmıştır.  $i$  ( $0 \leq i \leq n - 1$ ) numaralı düşman,  $i$  numaralı zindanda bulunmakta ve bu düşmanın gücü  $s[i]$  ile belirtilmektedir.  $n$  numaralı zindandaysa düşman bulunmuyor.

Kahramanımız zindanlara  $x$  numaralı zindandan  $z$  gücüyle giriş yapıyor. Ne zaman kahramanımız herhangi bir  $i$  ( $0 \leq i \leq n - 1$ ) numaralı zindana giriş yapsa, onu  $i$  numaralı düşman karşılıyor ve aşağıdakiler gerçekleşiyor:

- Eğer kahramanımızın gücü, düşmanın gücü olan  $s[i]$  değerinden büyük veya eşitse kahramanımız kazanıyor. Bunun neticesinde kahramanın gücü  $s[i]$  ( $s[i] \geq 1$ ) kadar **artıyor**. Bu durumda kahramanımız bir sonraki zindan olan  $w[i]$  ( $w[i] > i$ ) zindanına geçiyor.
- Aksi taktirde, kahramanımız kaybediyor. Bunun neticesinde kahramanın gücü  $p[i]$  ( $p[i] \geq 1$ ) kadar **artıyor**. Bu durumda kahramanımız bir sonraki zindan olan  $l[i]$  zindanına geçiyor.

$p[i]$  değeri  $s[i]$  değerinden küçük olabilir, aynı olabilir veya büyük olabilir. Aynı şekilde  $l[i]$  değeri de  $i$  değerinden küçük olabilir, aynı olabilir veya büyük olabilir. Çarpışmanın sonucundan bağımsız olarak  $i$  zindanındaki düşman  $s[i]$  değerindeki gücüyle aynı zindanda kalmaya devam edecektir.

Oyun, kahramanın  $n$  numaralı zindana girmesiyle son bulmaktadır. Oyuna ilgi duyan biri, kahramanın hangi zindandan hangi güçle başladığından bağımsız olarak, oyunun sonlu sayıda çarpışmayla biteceğini gösterebilir.

Robert sizden bu oyunu  $q$  adet simülasyon ile test etmenizi istiyor. Herbir simülasyon için Robert size başlangıç zindanı  $x$  ve başlangıç gücü  $z$  değerlerini veriyor. Sizin göreviniz, herbir simülasyon için oyun bittiğinde kahramanın gücünü bulmanız.

## Implementasyon Detayları

Aşağıda verilen fonksiyonları kodlamanız gerekmektedir:

```
void init(int n, int[] s, int[] p, int[] w, int[] l)
```

- $n$ : düşman sayısıdır.
- $s$ ,  $p$ ,  $w$ ,  $l$ : verilen  $n$  uzunluğunda diziler. Her  $0 \leq i \leq n - 1$  için:
  - $s[i]$  -  $i$  numaralı düşmanın gücü. Bu aynı zamanda kahramanın,  $i$  numaralı düşmanla yaptığı çarpışmayı kazandığı taktirde elde edeceği güç miktarıdır.
  - $p[i]$  - Kahramanın  $i$  numaralı düşman ile yapılan çarpışmayı kaybettiğinde elde edeceği güç miktarıdır.

- $w[i]$  - Kahramanın  $i$  numaralı düşman ile yapılan çarpışmayı kazandığında gideceği zindandır.
- $l[i]$  - Kahramanın  $i$  numaralı düşman ile yapılan çarpışmayı kaybettiğinde gideceği zindandır.
- Bu fonksiyon, aşağıda verilen `simulate` fonksiyonlarından önce tam olarak bir defa çağırılacaktır.

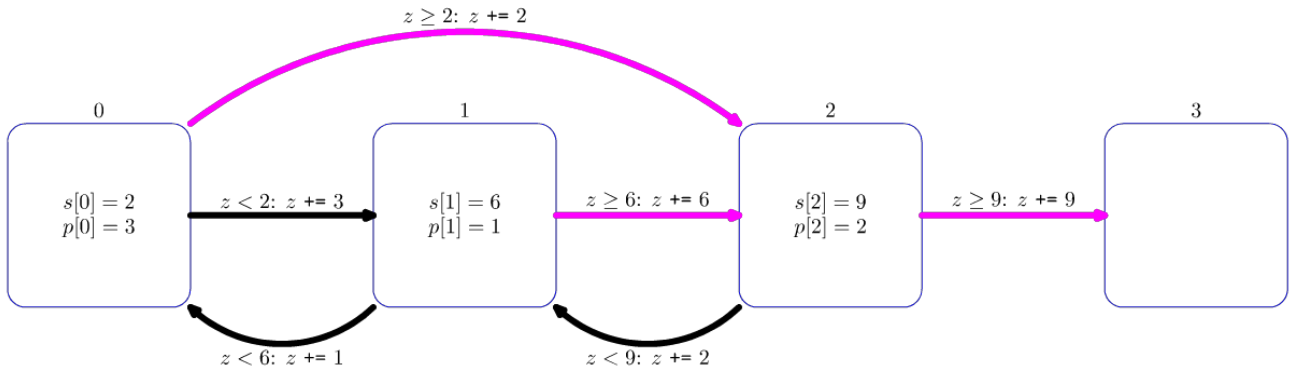
```
int64 simulate(int x, int z)
```

- $x$ : Kahramanın girdiği ilk zindan.
- $z$ : Kahramanın başlangıç gücü.
- Bu fonksiyon, oyun bittiğinde, kahramanın sahip olacağı güç değerini döndürmelidir. Kahramanımız oyuna  $x$  zindanında  $z$  güç değeriyle başlamaktadır.
- Bu fonksiyon tam olarak  $q$  defa çağırılacaktır.

## Örnekler

Aşağıda verilen fonksiyon çağırısı yapıldığında:

```
init(3, [2, 6, 9], [3, 1, 2], [2, 2, 3], [1, 0, 1])
```



Yukarıda verilen resim bu çağırışı göstermektedir, her kare bir zindanı temsil ediyor. Her zindan, 0, 1 and 2, için  $s[i]$  ve  $p[i]$  değerleri karelerin içinde verilmiştir. Mor çizgiler kahramanın çarpışmayı kazandıktan sonra geçtiği zindanı ve siyah çizgiler kahramanın çarpışmayı kaybettiğinde geçtiği zindanı göstermektedir.

Grader `simulate(0, 1)` fonksiyonunu çağırılmış olsun.

Oyun aşağıdaki şekilde devam edecektir:

Zindan	Kahramanın arpışma ncesindeki gc	Sonuç
0	1	Kaybetti
1	4	Kaybetti
0	5	Kazandı
2	7	Kaybetti
1	9	Kazandı
2	15	Kazandı
3	24	Oyun bitti

Bu durumda, fonksiyon ağrısı 24 deėerini dndrecektir.

Grader `simulate(2, 3)` fonksiyonunu aėırmış olsun.

Oyun ařaėıdaki řekilde devam edecektir:

Zindan	Kahramanın arpışma ncesindeki gc	Sonuç
2	3	Kaybetti
1	5	Kaybetti
0	6	Kazandı
2	8	Kaybetti
1	10	Kazandı
2	16	Kazandı
3	25	Oyun bitti

Bu durumda, fonksiyon ağrısı 25 deėerini dndrecektir.

## Kısıtlar

- $1 \leq n \leq 400\,000$
- $1 \leq q \leq 50\,000$
- $1 \leq s[i], p[i] \leq 10^7$  (tm  $0 \leq i \leq n - 1$  deėerleri iin)
- $0 \leq l[i], w[i] \leq n$  (tm  $0 \leq i \leq n - 1$  deėerleri iin)
- $w[i] > i$  (tm  $0 \leq i \leq n - 1$  deėerleri iin)
- $0 \leq x \leq n - 1$
- $1 \leq z \leq 10^7$

## Altėorevler

1. (11 puan)  $n \leq 50\,000$ ,  $q \leq 100$ ,  $s[i], p[i] \leq 10\,000$  (tm  $0 \leq i \leq n - 1$  deėerleri iin)

2. (26 puan)  $s[i] = p[i]$  (tüm  $0 \leq i \leq n - 1$  değerleri için)
3. (13 puan)  $n \leq 50\,000$ , tüm düşmanlar aynı güce sahip, başka bir deyişle, tüm  $0 \leq i, j \leq n - 1$  değerleri için  $s[i] = s[j]$ .
4. (12 puan)  $n \leq 50\,000$ , tüm  $s[i]$  değerleri için en fazla 5 adet farklı değer bulunmaktadır.
5. (27 puan)  $n \leq 50\,000$
6. (11 puan) Ek kısıt bulunmamaktadır.

## Örnek grader

Örnek grader girdiyi aşağıdaki formatta okuyacaktır:

- satır 1:  $n \ q$
- satır 2:  $s[0] \ s[1] \ \dots \ s[n - 1]$
- satır 3:  $p[0] \ p[1] \ \dots \ p[n - 1]$
- satır 4:  $w[0] \ w[1] \ \dots \ w[n - 1]$
- satır 5:  $l[0] \ l[1] \ \dots \ l[n - 1]$
- satır  $6 + i \ (0 \leq i \leq q - 1)$ :  $i$ 'ninci `simulate` çağrısı için  $x \ z$  değerleri

Örnek grader cevaplarınızı aşağıdaki formatta yazacaktır:

- line  $1 + i \ (0 \leq i \leq q - 1)$ :  $i$ 'ninci `simulate` çağrısı için döndürülen değer