

# Dungeons Game

Robert je dizajnira novu računarsku igricu. Igrica uključuje jednog junaka,  $n$  protivnika i  $n + 1$  ćelija (tamnica). Protivnici su numerisani od  $0$  do  $n - 1$  a ćelije su numerisane od  $0$  do  $n$ . Protivnik  $i$  ( $0 \leq i \leq n - 1$ ) se nalazi u ćeliji  $i$  i ima snagu  $s[i]$ . Nema ni jednog protivnika u posljednjoj ćeliji  $n$ .

Naš junak započinje ulaskom u ćeliju  $x$ , i raspolaže sa snagom  $z$ . Prilikom svakog ulaska u neku ćeliju  $i$  ( $0 \leq i \leq n - 1$ ), junak se suočava i sukobljava sa protivnikom  $i$ , i to rezultira sa jednim od sljedeća dva ishoda:

- Ako je snaga junaka veća ili jednaka snazi protivnika u ćeliji  $s[i]$ , junak pobjeđuje. Ovo znači da će se snaga junaka **povećati** za  $s[i]$  ( $s[i] \geq 1$ ). U ovom slučaju, sljedeća ćelija u koju junak ulazi je  $w[i]$  gdje je  $w[i] > i$ .
- U suprotnom, naš junak gubi. Ovo rezultira time da se snaga junaka **poveća** za  $p[i]$  ( $p[i] \geq 1$ ). U ovom slučaju, sljedeća ćelija u koju junak ulazi je  $l[i]$ .

Primjetimo da  $p[i]$  može biti manja, jednaka ili veća od  $s[i]$ . Na isti način,  $l[i]$  može biti manja, jednaka ili veća od  $i$ . Bez obzira na rezultat sukoba u ćeliji, protivnik ostaje u svojoj ćeliji  $i$  i zadržava snagu  $s[i]$ .

Igra završava kada junak uđe u ćeliju broj  $n$ . Moguće je pokazati da igra završava nakon konačnog broja sukobljavanja, bez obzira na početnu ćeliju koju će posjetiti junak i bez obzira na njegovu početnu snagu.

Robert traži od vas da testirate njegovu novu igricu izvršavajući  $q$  simulacija. Za svaku simulaciju, Robert će definisati početnu ćeliju  $x$  koju treba posjetiti i veličinu početne snage  $z$ . Vaš zadatak je da otkrijete, nakon svake simulacije, snagu junaka nakon što igrica završi.

## Detalji implementacije

Potrebno je implementirati sljedeću proceduru:

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

- $n$ : ukupan broj protivnika.
- $s, p, w, l$ : nizovi dužine  $n$ . Za  $0 \leq i \leq n - 1$ :
  - $s[i]$  je snaga protivnika  $i$ . To je takođe dodatna snaga koju će dobit naš junak ako pobjedi protivnika  $i$ .
  - $p[i]$  je dodatna snaga koju će dobit naš junak nakon što izgubi od protivnika  $i$ .
  - $w[i]$  je ćelija u koju će junak ući nakon što pobjedi protivnika  $i$ .
  - $l[i]$  je ćelija u koju će junak ući nakon što izgubi protivnika  $i$ .

- Ova procedura se poziva samo jednom, prije bilo kojeg poziva procedure `simulate` (pogledati detalje dolje).

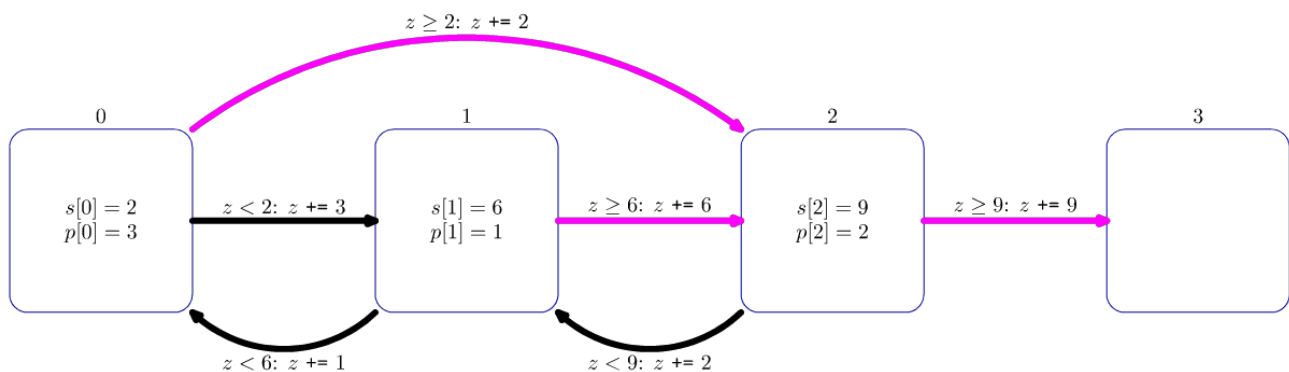
```
int64 simulate(int x, int z)
```

- $x$ : prva, početna ćelija u koju junak ulazi.
- $z$ : početna snaga koju junak posjeduje.
- Ova procedura treba da vrati snagu koju posjeduje junak u momentu kada igra završi, pod uslovom da junak igru započinje sa ćelijom  $x$ , i sa početnom snagom  $z$ .
- Ova procedura se poziva tačno  $q$  puta.

## Primjer

Razmotrimo sljedeći poziv :

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



Dijagram slikovito prikazuje šta se dešava tokom ovog poziva. Svaki kvadrat predstavlja jednu ćeliju. Za ćelije 0, 1 i 2, vrijednosti  $s[i]$  i  $p[i]$  su date unutar odgovarajućih kvadrata. Strelice boje magenta prikazuju gdje junak odlazi nakon što pobjedi u sukobu sa protivnikom a crne strelice pokazuju gdje junak odlazi nakon što izgubi u tom sukobu.

Recimo da sada grader poziva `simulate(0, 1)`.

Igrica se odvija na sljedeći način:

Ćelija	Snaga junaka prije sukoba	Rezult
0	1	Gubi
1	4	Gubi
0	5	Pobjeđuje
2	7	Gubi
1	9	Pobjeđuje
2	15	Pobjeđuje
3	24	Igra završava

Prema tome, procedura treba da vrati broj 24.

Recimo da sada grader poziva `simulate(2, 3)`.

Igrica se odvija na sljedeći način:

Tamnica	Snaga junaka prije sukoba	Rezult
2	3	Gubi
1	5	Gubi
0	6	Pobjeđuje
2	8	Gubi
1	10	Pobjeđuje
2	16	Pobjeđuje
3	25	Igra završava

Prema tome, procedura treba da vrati broj 25.

## Ograničenja

- $1 \leq n \leq 400\,000$
- $1 \leq q \leq 50\,000$
- $1 \leq s[i], p[i] \leq 10^7$  (za sve  $0 \leq i \leq n - 1$ )
- $0 \leq l[i], w[i] \leq n$  (za sve  $0 \leq i \leq n - 1$ )
- $w[i] > i$  (za sve  $0 \leq i \leq n - 1$ )
- $0 \leq x \leq n - 1$
- $1 \leq z \leq 10^7$

## Podzadaci

1. (11 bodova)  $n \leq 50\,000$ ,  $q \leq 100$ ,  $s[i], p[i] \leq 10\,000$  (za sve  $0 \leq i \leq n - 1$ )

2. (26 bodova)  $s[i] = p[i]$  (za sve  $0 \leq i \leq n - 1$ )
3. (13 bodova)  $n \leq 50\,000$ , svi protivnici imaju istu snagu, drugim riječima,  $s[i] = s[j]$  za sve  $0 \leq i, j \leq n - 1$ .
4. (12 bodova)  $n \leq 50\,000$ , ima najviše 5 različitih vrijednosti među svim vrijednostima  $s[i]$ .
5. (27 bodova)  $n \leq 50\,000$
6. (11 bodova) Nema dodatnih ograničenja

## Sample grader

The sample grader čita na ulazu podatku u sljedećem formatu:

- linija 1:  $n \ q$
- linija 2:  $s[0] \ s[1] \ \dots \ s[n - 1]$
- linija 3:  $p[0] \ p[1] \ \dots \ p[n - 1]$
- linija 4:  $w[0] \ w[1] \ \dots \ w[n - 1]$
- linija 5:  $l[0] \ l[1] \ \dots \ l[n - 1]$
- linija  $6 + i$  ( $0 \leq i \leq q - 1$ ):  $x \ z$  za  $i$ -ti poziv procedure `simulate`.

Sample grader štampa vaše odgovore u sljedećem formatu:

- linija  $1 + i$  ( $0 \leq i \leq q - 1$ ): vrijednost koju je vratila procedura `simulate` prilikom  $i$ -tog poziva.