

# Dungeons Game

Robert ontwerpt een nieuw spel. Het spel heeft één held,  $n$  tegenstanders en  $n + 1$  kerkers. De tegenstanders zijn genummerd van  $0$  tot en met  $n - 1$  en de kerkers van  $0$  tot en met  $n$ . Tegenstander  $i$  ( $0 \leq i \leq n - 1$ ) staat in kerker  $i$  en heeft kracht  $s[i]$ . Er is geen tegenstander in kerker  $n$ .

De held begint in kerker  $x$ , met kracht  $z$ . Iedere keer dat de held een kerker  $i$  ( $0 \leq i \leq n - 1$ ) betreedt, confronteert hij tegenstander  $i$  en gebeurt een van de volgende dingen:

- Als de kracht van de held groter dan of gelijk is aan de kracht van tegenstander  $s[i]$ , dan wint de held. Hierdoor **neemt de kracht van de held toe** met  $s[i]$  ( $s[i] \geq 1$ ). In dit geval gaat de held vervolgens naar kerker  $w[i]$  ( $w[i] > i$ ).
- Zo niet, dan verliest de held. Hierdoor **neemt de kracht van de held toe** met  $p[i]$  ( $p[i] \geq 1$ ). In dit geval gaat de held vervolgens naar kerker  $l[i]$ .

Merk op dat  $p[i]$  kleiner dan, gelijk aan of groter dan  $s[i]$  kan zijn.  $l[i]$  kan ook kleiner dan, gelijk aan, of groter dan  $i$  zijn.

Ongeacht het resultaat van de confrontatie blijft de tegenstander in kerker  $i$  en behoudt kracht  $s[i]$ .

Het spel is afgelopen als de held kerker  $n$  betreedt. Het valt aan te tonen dat het spel na een eindig aantal confrontaties afgelopen is, ongeacht in welke kerker de held start en met welke kracht.

Robert vraagt je om zijn spel te testen door  $q$  simulaties te draaien. Voor elke simulatie bepaalt Robert de start kerker  $x$  en start kracht  $z$ . Jouw taak is om voor iedere simulatie de kracht van de held aan het einde van het spel te bepalen.

## Implementatiedetails

Implementeer de volgende functies:

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

- $n$ : aantal tegenstanders.
- $s$ ,  $p$ ,  $w$ ,  $l$ : arrays van lengte  $n$ . Voor  $0 \leq i \leq n - 1$ :
  - $s[i]$  is de kracht van tegenstander  $i$ . Het is ook de kracht die de held erbij krijgt als hij van tegenstander  $i$  wint.
  - $p[i]$  is de kracht die de held erbij krijgt als hij verliest van tegenstander  $i$ .
  - $w[i]$  is de kerker waar de held heen gaat als hij wint van tegenstander  $i$ .
  - $l[i]$  is de kerker waar de held heen gaat als hij verliest van tegenstander  $i$ .

- Deze functie wordt precies één keer aangeroepen, voordat `simulate` wordt aangeroepen (zie hieronder).

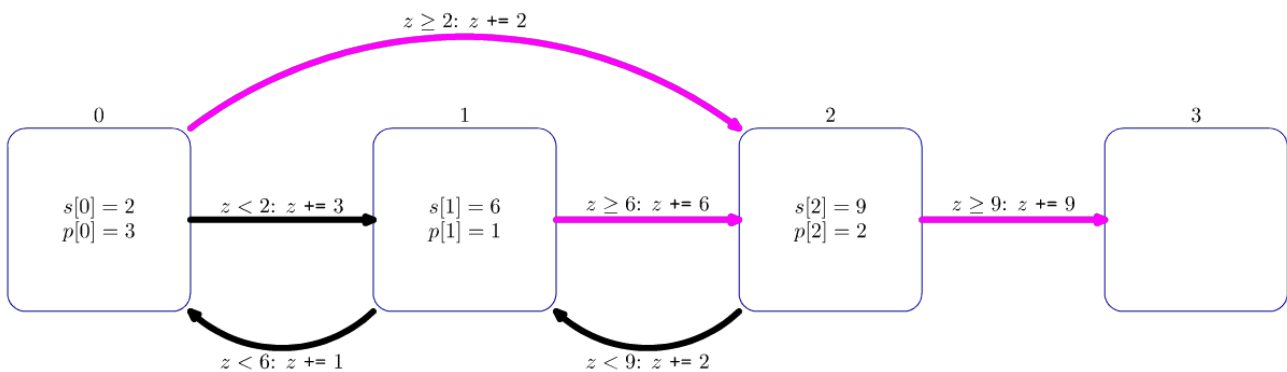
```
int64 simulate(int x, int z)
```

- $x$ : de kerker waar de held begint.
- $z$ : de kracht waar de held mee begint.
- Deze functie moet de kracht van de held teruggeven nadat het spel is afgelopen, wanneer de held in kerker  $x$  begint met kracht  $z$ .
- Deze functie wordt precies  $q$  keer aangeroepen.

## Voorbeeld

Neem de volgende aanroep:

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



Het bovenstaande diagram illustreert deze aanroep. Ieder vakje is een kerker. Voor kerkers 0, 1 en 2 zijn de waarden  $s[i]$  en  $p[i]$  aangegeven in de vakjes. De rode pijlen geven aan waar de held heen gaat als hij een confrontatie wint en de zwarte pijlen als hij een confrontatie verliest.

Stel dat de grader `simulate(0, 1)` aanroept.

Het spel verloopt dan als volgt:

Kerker	Kracht voor de confrontatie	Resultaat
0	1	Verlies
1	4	Verlies
0	5	Win
2	7	Verlies
1	9	Win
2	15	Win
3	24	Spel stopt

De functie moet dus 24 teruggeven.

Stel dat de grader `simulate(2, 3)` aanroept.

Het spel loopt als volgt:

Kerker	Kracht voor de confrontatie	Resultaat
2	3	Verlies
1	5	Verlies
0	6	Win
2	8	Verlies
1	10	Win
2	16	Win
3	25	Spel stopt

De functie moet dus 25 teruggeven.

## Randvoorwaarden

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

## Subtasks

1. (11 punten)  $n \leq 50\,000$ ,  $q \leq 100$ ,  $s[i], p[i] \leq 10\,000$  (voor iedere  $0 \leq i \leq n - 1$ )
2. (26 punten)  $s[i] = p[i]$  (voor iedere  $0 \leq i \leq n - 1$ )
3. (13 punten)  $n \leq 50\,000$ , alle tegenstanders hebben dezelfde kracht, in andere woorden,  $s[i] = s[j]$  voor iedere  $0 \leq i, j \leq n - 1$ .
4. (12 punten)  $n \leq 50\,000$ , er zijn hoogstens 5 verschillende waarden tussen alle waarden van  $s[i]$ .
5. (27 punten)  $n \leq 50\,000$
6. (11 punten) Geen aanvullende randvoorwaarden.

## Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat:

- regel 1:  $n \ q$

- regel 2:  $s[0] \ s[1] \ \dots \ s[n-1]$
- regel 3:  $p[0] \ p[1] \ \dots \ p[n-1]$
- regel 4:  $w[0] \ w[1] \ \dots \ w[n-1]$
- regel 5:  $l[0] \ l[1] \ \dots \ l[n-1]$
- regel  $6 + i$  ( $0 \leq i \leq q-1$ ):  $x \ z$  voor aanroep  $i$  aan `simulate`.

De voorbeeldgrader schrijft je antwoorden in het volgende formaat:

- regel  $1 + i$  ( $0 \leq i \leq q-1$ ): de waarde teruggekregen van aanroep  $i$  aan `simulate`.