

Dungeons Game

Роберт разработва нова компютърна игра. Играта включва един герой, n опоненти и $n + 1$ подземия. Опонентите са номерирани с числата от 0 до $n - 1$, а подземията са номерирани с числата от 0 до n . Опонент i ($0 \leq i \leq n - 1$) се намира в подземие i и има сила $s[i]$. Няма опонент в подземие n .

Героят започва като влиза в подземие x със сила z . Всеки път, когато той влезе в някое подземие i ($0 \leq i \leq n - 1$), той се изправя срещу опонент i , и се случва точно един от следните сценарии:

- Ако силата на героя е по-голяма или равна на силата на опонента $s[i]$, то героят печели. Като следствие от това, силата на героя се **увеличава** със $s[i]$ ($s[i] \geq 1$). В този случай героят влиза в подземие $w[i]$ ($w[i] > i$).
- В противен случай, героят губи. Като следствие от това, силата на героя се **увеличава** с $p[i]$ ($p[i] \geq 1$). В този случай героят влиза в подземие $l[i]$.

Обърнете внимание, че $p[i]$ може да бъде по-малко, равно или по-голямо от $s[i]$. Също $l[i]$ може да бъде по-малко, равно или по-голямо от i . Без значение от резултата на битката, опонентът остава в подземие i и запазва своята сила $s[i]$.

Играта приключва, когато героят влезе в последното подземие n . Може да се покаже, че играта свършва след краен брой битки, без значение от началното подземие на героя и началната му сила.

Роберт иска от Вас да тествата неговата игра, като изпълните q симулации. За всяка симулация, Роберт определя началното подземие x и началната сила z . Вашата задача е за всяка симулация да откриете силата на героя, когато играта завърши.

Детайли по реализацията

Трябва да напишете следната функция:

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

- n : брой на опонентите.
- s , p , w , l : масиви с дължина n . За всяко $0 \leq i \leq n - 1$:
 - $s[i]$ е силата на опонент i . Също така това е силата, която получава героя, след като спечели срещу опонент i .
 - $p[i]$ е силата, която получава героя, след като загуби срещу опонент i .
 - $w[i]$ е подземиято, в което влиза героя, след като спечели срещу опонент i .

- $l[i]$ е подземие, в което влиза героя, след като загуби срещу опонент i .
- Тази функция се вика само веднъж, преди извикванията на функцията `simulate` (вижте следващия абзац).

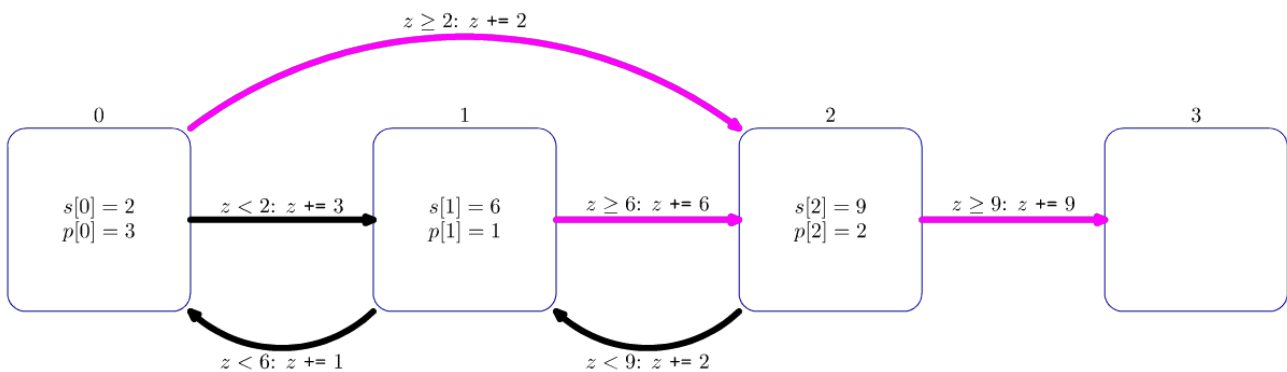
```
int64 simulate(int x, int z)
```

- x : начаното подземие, в което влиза героя.
- z : началната сила на героя.
- Тази функция трябва да върне силата на героя, когато играта завърши, при условие, че героят започва с влизане в подземие x и начална сила z .
- Тази функция се вика точно q пъти.

Пример

Нека имаме следното извикване:

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



Горната диаграма изобразява данните от това извикване. Всеки квадрат съответства на подземие. За подземия 0, 1 и 2, стойностите $s[i]$ и $p[i]$ са указани в квадратите. Лилавите стрелки указват къде се придвижва героя, след като спечели битка, докато черните стрелки указват къде се придвижва след загуба.

Нека грейдърът направи извикване `simulate(0, 1)`.

Играта протича по следния начин:

Подземие	Сила на героя преди битката	Резултат
0	1	Загуба
1	4	Загуба
0	5	Победа
2	7	Загуба
1	9	Победа
2	15	Победа
3	24	Играта завършва

Така функцията трябва да върне 24 в този случай.

Нека грейдърът направи извикване `simulate(2, 3)`.

Играта протича по следния начин:

Подземие	Сила на героя преди битката	Резултата
2	3	Загуба
1	5	Загуба
0	6	Победа
2	8	Загуба
1	10	Победа
2	16	Победа
3	25	Играта завършва

Така функцията трябва да върне 25 в този случай.

Ограничения

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

Подзадачи

1. (11 точки) $n \leq 50\,000$, $q \leq 100$, $s[i], p[i] \leq 10\,000$ (за всяко $0 \leq i \leq n - 1$)

2. (26 точки) $s[i] = p[i]$ (за всяко $0 \leq i \leq n - 1$)
3. (13 точки) $n \leq 50\,000$, всички опоненти имат равна сила, с други думи $s[i] = s[j]$ за всяко $0 \leq i, j \leq n - 1$.
4. (12 точки) $n \leq 50\,000$, има най-много 5 уникални стойности сред всички стойности на $s[i]$.
5. (27 точки) $n \leq 50\,000$
6. (11 точки) няма допълнителни ограничения.

Примерен грейдър

Примерният грейдър чете от стандартния вход в следния формат:

- ред 1: $n \ q$
- ред 2: $s[0] \ s[1] \ \dots \ s[n - 1]$
- ред 3: $p[0] \ p[1] \ \dots \ p[n - 1]$
- ред 4: $w[0] \ w[1] \ \dots \ w[n - 1]$
- ред 5: $l[0] \ l[1] \ \dots \ l[n - 1]$
- ред $6 + i$ ($0 \leq i \leq q - 1$): $x \ z$ за i -тото извикване на функцията `simulate`.

Примерният грейдър отпечатва вашите отговори в следния формат:

- ред $1 + i$ ($0 \leq i \leq q - 1$): върната стойност от i -тото извикване на функцията `simulate`.