

მინისქვეშა თამაში

რობერტი ჰქმნის ახალ კომპიუტერულ თამაშს. თამაშში მონაწილეობს ერთი გმირი, n ოპონენტი და $n + 1$ მინისქვეშეთი. ოპონენტები გადანომრილია 0-დან $(n - 1)$ -მდე და მინისქვეშეთები გადანომრილია 0-დან n -მდე. i -ური ($0 \leq i \leq n - 1$) ოპონენტი იმყოფება i -ურ მინისქვეშეთში და აქვს $s[i]$ ძალა. n ნომრის მქონე მინისქვეშეთში ოპონენტი არ არის.

გმირი სტარტს იღებს x ნომრის მქონე მინისქვეშეთიდან და მას აქვს ძალა z . ყოველთვის, როცა გმირი ჩადის i -ურ ($0 \leq i \leq n - 1$) მინისქვეშეთში, მას ებრძვის i -ური ოპონენტი და ხდება შემდეგი შემთხვევებიდან ერთ-ერთი:

- თუ გმირის ძალა მეტია ან ტოლია ოპონენტის $s[i]$ ძალაზე, მაშინ გმირი იმარჯვებს. ამ შემთხვევაში გმირის ძალა იზრდება $s[i]$ ($s[i] \geq 1$ ერთეულით და გმირი გადადის $w[i]$ ($w[i] > i$) მინისქვეშეთში.
- წინააღმდეგ შემთხვევაში გმირი მარცხდება. ამ დროს გმირის ძალა იზრდება $p[i]$ ($p[i] \geq 1$) ერთეულით და გმირი გადადის $l[i]$ მინისქვეშეთში.

შენიშვნა. $p[i]$ შეიძლება იყოს ნაკლები, ტოლი ან მეტი $s[i]$ -ზე. ასევე, $l[i]$ შეიძლება იყოს ნაკლები, ტოლი ან მეტი i -ზე. ორთაბრძოლის შედეგის მიუხედავად ოპონენტი რჩება i -ურ მინისქვეშეთში და ინარჩუნებს $s[i]$ ძალას.

თამაში მთავრდება, როდესაც გმირი მიაღწევს n ნომრის მქონე მინისქვეშეთს. შესაძლებელია იმის ჩვენება, რომ თამაში დასრულდება ორთაბრძოლების სასრული რიცხვის შემდეგ საწყისი მინისქვეშეთის და ძალის მიუხედავად.

რობერტი გთხოვთ, გატესტოთ მისი თამაში q სიმულაციისათვის. ყოველი სიმულაციისას, რობერტი განსაზღვრავს საწყის x მინისქვეშეთს და საწყის z ძალას. თქვენი ამოცანაა ყოველი სიმულაციისათვის განსაზღვროთ გმირის ძალა თამაშის დასრულების შემდეგ.

იმპლემენტაციის დეტალები

თქვენ უნდა მოახდინოთ შემდეგი პროცედურის იმპლემენტაცია:

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

- n : ოპონენტთა რაოდენობა.
- s , p , w , l : n სიგრძის მასივები. $0 \leq i \leq n - 1$:
 - $s[i]$ წარმოადგენს i -ური ოპონენტის ძალას. ასევე ესაა ძალა, რომელსაც დაემატება გმირის ძალას i ოპონენტთან გამარჯვების შემთხვევაში.
 - $p[i]$ წარმოადგენს ძალას, რომიც დაემატება გმირის ძალას i -ურ ოპონენტთან დამარცხების შემთხვევაში.

- $w[i]$ წარმოადგენს მინისქვეშეთს, რომელშიც გადავა გმირი i -ური ოპონენტთან გამარჯვების შემდეგ.
- $l[i]$ წარმოადგენს მინისქვეშეთს, რომელშიც გადავა გმირი i -ური ოპონენტთან დამარცხების შემდეგ.
- ეს ფუნქცია გამოიძახება ზუსტად ერთხელ, ფუნქცია `simulate` გამოიძახებამდე.

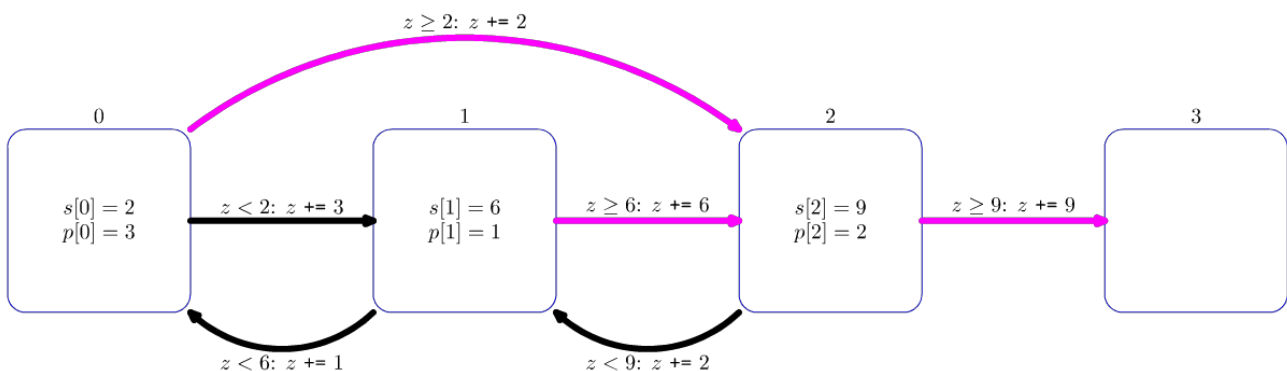
```
int64 simulate(int x, int z)
```

- x : საწყისი მინისქვეშეთი, სადაც შედის გმირი.
- z : გმირის საწყისი ძალა.
- ამ ფუნქციამ უნდა დააბრუნოს გმირის ძალა თამაშის დასრულების შემდეგ, იმის გათვალისწინებით, რომ მან დაიწყო x ნომრის მქონე მინისქვეშეთიდან, საწყისი z ძალით.
- ფუნქცია გამოიძახება ზუსტად q -ჯერ.

მაგალითი

განვიხილოთ შემდეგი გამოძახება:

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



დიაგრამა აღწერს ამ გამოძახებას. ყოველი კვადრეტი აღწერს მინისქვეშეთს. 0, 1 და 2 მინისქვეშეთებისთვის, $s[i]$ და $p[i]$ ჩანერილია კვადრატის შიგნით. იასამნისფერი ისრები მიუთითებენ საით წავა გმირი მოგების შემდეგ, ხოლო შავი ისრები მიუთითებენ, საით წავა გმირი წაგების შემდეგ.

ვთქვათ, გრადერი იძახებს `simulate(0, 1)`.

თამაში წარიმართება შემდეგნაირად:

მინისქვეშეთი	გმირის ძალა ბრძოლის წინ	შედეგი
0	1	წაგება
1	4	წაგება
0	5	მოგება
2	7	წაგება
1	9	მოგება
2	15	მოგება
3	24	დასასრული

ამრიგად, ფუნქციამ უნდა დააბრუნოს 24.

ვთქვათ, გრადერი იძახებს `simulate(2, 3)`.

თამაში წარიმართება შემდეგნაირად:

მინისქვეშეთი	გმირის ძალა ბრძოლის წინ	შედეგი
2	3	წაგება
1	5	წაგება
0	6	მოგება
2	8	წაგება
1	10	მოგება
2	16	მოგება
3	25	დასასრული

ამრიგად, ფუნქციამ უნდა დააბრუნოს 25.

შეზღუდვები

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

ქვეამოცანები

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

2. (26 ქულა) $s[i] = p[i]$ (ყველა $0 \leq i \leq n - 1$)
3. (13 ქულა) $n \leq 50\,000$, ყველა ოპონენტს ერთნაირი ძალა აქვს, ანუ $s[i] = s[j]$ ყველა $0 \leq i, j \leq n - 1$.
4. (12 ქულა) $n \leq 50\,000$, $s[i]$ -ების მნიშვნელობებს შორის არაუმეტეს 5 განსხვავებული მნიშვნელობაა.
5. (27 ქულა) $n \leq 50\,000$
6. (11 ქულა) დამატებითი შეზღუდვების გარეშე.

სანიმუშო გრადერი

სანიმუშო გრადერს შეაქვს მონაცემები შემდეგი ფორმატით:

- სტრიქონი 1: $n \ q$
- სტრიქონი 2: $s[0] \ s[1] \ \dots \ s[n - 1]$
- სტრიქონი 3: $p[0] \ p[1] \ \dots \ p[n - 1]$
- სტრიქონი 4: $w[0] \ w[1] \ \dots \ w[n - 1]$
- სტრიქონი 5: $l[0] \ l[1] \ \dots \ l[n - 1]$
- სტრიქონი $6 + i$ ($0 \leq i \leq q - 1$): $x \ z \ \text{simulate}$ -ს i -ური გამოძახებისთვის.

სანიმუშო გრადერს გამოაქვს მონაცემები შემდეგი ფორმატით:

- სტრიქონი $1 + i$ ($0 \leq i \leq q - 1$): აბრუნებს ფუნქცია `simulate`-ს i -ური გამოძახების მნიშვნელობას.