Гра у підземеллі

Роман розробляє нову комп'ютерну гру. У грі присутній один герой, n супротивників та $n+1$ підзмель. Супротивники пронумеровані від 0 до $n-1$, а підземелля пронумеровані від 0 до n. Супротивник i ($0 \leq i \leq n-1$) знаходиться у підземеллі i і має силу $s[i]$. У підземеллі n супротивника немає.

Герой розпочинає гру заходом до підземелля x, з силою z. Кожен раз коли герой вхдить до підземелля i ($0 \leq i \leq n-1$), він змагається з супротивником i, і можливі наступні сценарії:

- Якщо сила героя більша, або рівна за силу супротивника $s[i]$, герой перемагає. При цьому сила героя **збільшується** на $s[i]$ ($s[i] \geq 1$). При цьому герой переходить до підземелля $w[i]$ ($w[i] > i$).
- Інакше герой програє. При цьому сила героя **збільшується** на $p[i]$ ($p[i] \geq 1$). В такому випадку герой переходить до підземелля $l[i]$.

Зауважимо, що $p[i]$ може бути меншим, рівним, або більшим ніж $s[i]$.Також, $l[i]$ може бути меншим, рівним, або більшим ніж i. Незалежно від того, чим закінчилось протистояння, супротивник залишається в підземеллі i і його сила залишається рівною $s[i]$.

Гра закінчується, коли герой потрапляє до підземелля n. Очевидно, що гра закінчиться за скінченну кількість кроків незалежно від початкової сили героя та стартового підземелля.

Роман просить вас протестувати гру в ході q симуляцій. Для кожної симуляції Роман визначає стартове підземелля x та стартову силу z. Ваша задача - визначити для кожної симуляції силу героя на момент завершення гри.

Деталі реалізації

Вам потрібно реалізувати наступні процедури:

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

- n: кількість супротивників.
- s, p, w, l: масиви довжини n. Для $0 \leq i \leq n-1$:
  - $s[i]$ сила супротивника i. Відповідно це число, на яке збільшується сила героя у випадку перемоги над супротивником i.
  - $p[i]$ число, на яке збільшується сила героя у випадку програшу супротивнику i.
  - $w[i]$ номер підземелля куди переходить герой у випадку перемоги над супротивником i.
  - $l[i]$ номер підземелля куди переходить герой у випадку поразки супротивнику i.

- Ця процедура викликається рівно 1 раз перед викликами процедури `simulate` (дивись далі).

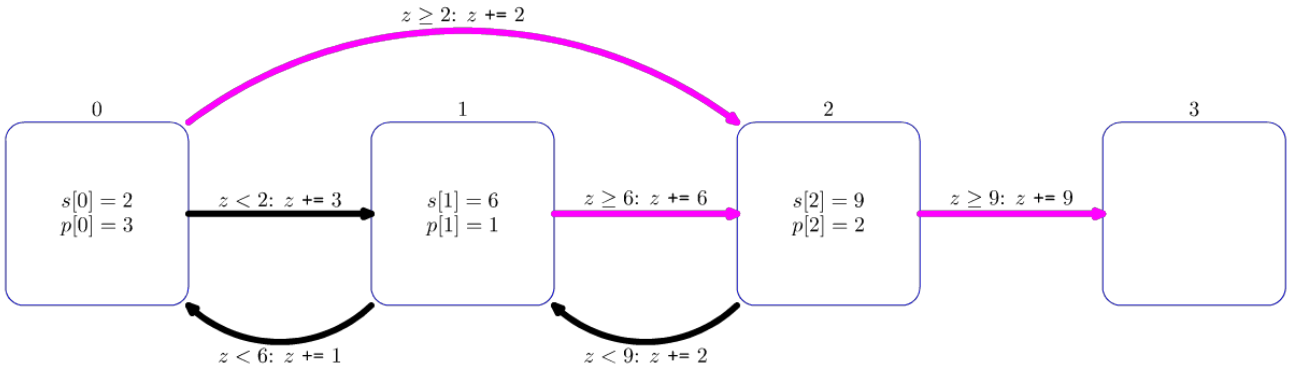
```
int64 simulate(int x, int z)
```

- x: початкове підземелля.
- z: початкова сила героя.
- Процедура повинна повернути силу героя на момент завершення гри, припускаючи, що герой розпочинає гру у підземеллі x, з силою z.
- Процедура викликається рівно q разів.

Приклад

Розглянемо наступний виклик:

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



Діаграма вище описує виклик. Кожен квадрат представляє підземелля. Для підземель 0, 1 та 2, значення $s[i]$ та $p[i]$ показані всередині квадратів. Рожеві стрілки показуть куди перейде герой у випадку перемоги, а чорні - у випадку поразки.

Припусимо, що модуль перевірки робить виклик `simulate(0, 1)`.

Гра піде за наступним сценарієм:

Підземелля	Сила героя до протистояння	Результат
0	1	поразка
1	4	поразка
0	5	перемога
2	7	поразка
1	9	перемога
2	15	перемога
3	24	кінець гри

Отже процедура повинна повернути значення 24.

Припусимо, що модуль перевірки робить виклик `simulate(2, 3)`.

Гра піде за наступним сценарієм:

Підземелля	Сила героя до протистояння	Результат
2	3	поразка
1	5	поразка
0	6	перемога
2	8	поразка
1	10	перемога
2	16	перемога
3	25	кінець гри

Отже процедура повинна повернути значення 25.

Обмеження

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

Підзадачі

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

2. (26 балів) $s[i] = p[i]$ (для всіх $0 \le i \le n-1$)
3. (13 балів) $n \le 50\,000$, у всіх супротивників однакова сила, іншими словами $s[i] = s[j]$ для всіх $0 \le i, j \le n-1$.
4. (12 балів) $n \le 50\,000$, існує не більше 5 різних значень сили $s[i]$.
5. (27 балів) $n \le 50\,000$
6. (11 балів) Без додаткових обмежень.

Приклад модуля перевірки

Модуль перевірки зчитує дані у наступному форматі:

- рядок 1: n q
- рядок 2: $s[0]$ $s[1]$ $\ldots$ $s[n-1]$
- рядок 3: $p[0]$ $p[1]$ $\ldots$ $p[n-1]$
- рядок 4: $w[0]$ $w[1]$ $\ldots$ $w[n-1]$
- рядок 5: $l[0]$ $l[1]$ $\ldots$ $l[n-1]$
- рядок $6+i$ ($0 \le i \le q-1$): x z для i-го виклику `simulate`.

Модуль перевірки виводить дані у наступному форматі:

- рядок $1+i$ ($0 \le i \le q-1$) : значення яке повернув i-й виклик `simulate`.