

# Игра в подземелья

Роберт разрабатывает новую компьютерную игру. В игре участвует один герой,  $n$  врагов и  $n + 1$  подземелье. Враги пронумерованы от 0 до  $n - 1$ , подземелья пронумерованы от 0 до  $n$ . Враг  $i$  ( $0 \leq i \leq n - 1$ ) находится в подземелье  $i$  и имеет силу  $s[i]$ . В подземелье  $n$  врагов нет.

Герой начинает с подземелья  $x$ , с начальной силой  $z$ . Каждый раз, когда герой попадает в какое-то подземелье  $i$  ( $0 \leq i \leq n - 1$ ), он сражается с врагом  $i$ , и происходит одно из следующих событий:

- Если сила героя больше или равна силе врага  $s[i]$ , герой побеждает. Это приводит к **увеличению** силы героя на  $s[i]$  ( $s[i] \geq 1$ ). В этом случае герой переходит в подземелье  $w[i]$  ( $w[i] > i$ ).
- В противном случае герой проигрывает. Это приводит к **увеличению** силы героя на  $p[i]$  ( $p[i] \geq 1$ ). В этом случае герой переходит в подземелье  $l[i]$ .

Заметим, что  $p[i]$  может быть меньше, равно или больше, чем  $s[i]$ . Также,  $l[i]$  может быть меньше, равно или больше, чем  $i$ . Независимо от исхода сражения, враг остаётся в подземелье  $i$ , и сохраняет силу  $s[i]$ .

Игра заканчивается, когда герой попадает в подземелье  $n$ . Можно показать, что игра заканчивается через конечное число сражений, независимо от стартового подземелья и силы героя.

Роберт просит Вас протестировать его игру, проведя  $q$  симуляций. Для каждой симуляции Роберт определил стартовое подземелье  $x$  и стартовую силу  $z$ . Ваша задача выяснить силу героя в конце игры для каждой симуляции.

## Детали реализации

Вы должны реализовать следующие функции:

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

- $n$ : количество врагов.
- $s$ ,  $p$ ,  $w$ ,  $l$ : массивы длины  $n$ . Для  $0 \leq i \leq n - 1$ :
  - $s[i]$  это сила врага  $i$ , а также сила, получаемая героем после победы над врагом  $i$ .
  - $p[i]$  это сила, получаемая героем после поражения от врага  $i$ .
  - $w[i]$  это подземелье, куда переходит герой после победы над врагом  $i$ .
  - $l[i]$  это подземелье, куда переходит герой после поражения от врага  $i$ .
- Эта функция вызывается ровно один раз, до вызовов `simulate` (смотри ниже).

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

- $x$ : стартовое подземелье для героя.
- $z$ : стартовая сила героя.
- Эта функция должна вернуть силу героя в конце игры, если герой начинает игру с подземелья  $x$ , имея силу  $z$ .
- Эта функция будет вызвана ровно  $q$  раз.

## Пример

Рассмотрим следующий вызов:

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

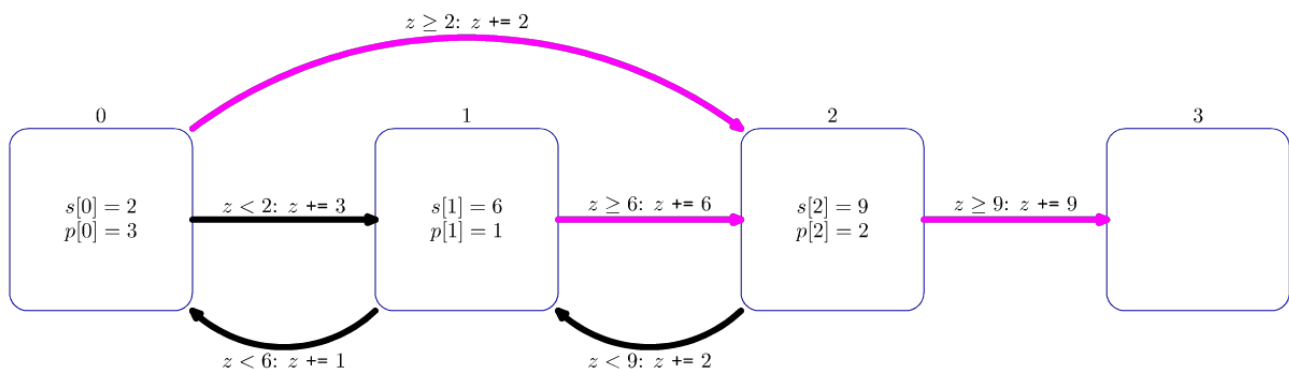


Диаграмма выше соответствует этому вызову. Каждый квадрат обозначает подземелье. Для подземелий 0, 1 и 2, значения  $s[i]$  и  $p[i]$  записаны внутри квадратов. Пурпурные стрелки показывают, куда герой перемещается в случае победы, а чёрные — куда герой перемещается в случае поражения.

Пусть грейдер вызвал функцию `simulate(0, 1)`.

Игра пройдет следующим образом:

Подземелье	Сила героя до сражения	Результат
0	1	Поражение
1	4	Поражение
0	5	Победа
2	7	Поражение
1	9	Победа
2	15	Победа
3	24	Конец игры

Таким образом, функция должна вернуть 24.

Пусть грейдер вызвал функцию `simulate(2, 3)`.

Игра пройдет следующим образом:

Подземелье	Сила героя до сражения	Результат
2	3	Поражение
1	5	Поражение
0	6	Победа
2	8	Поражение
1	10	Победа
2	16	Победа
3	25	Конец игры

Таким образом, функция должна вернуть 25.

## Ограничения

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

## Подзадачи

1. (11 баллов)  $n \leq 50\,000$ ,  $q \leq 100$ ,  $s[i], p[i] \leq 10\,000$  (для всех  $0 \leq i \leq n - 1$ )
2. (26 баллов)  $s[i] = p[i]$  (для всех  $0 \leq i \leq n - 1$ )
3. (13 баллов)  $n \leq 50\,000$ , все враги имеют одинаковую силу, другими словами,  $s[i] = s[j]$  для всех  $0 \leq i, j \leq n - 1$ .
4. (12 баллов)  $n \leq 50\,000$ , есть не более 5 различных значений среди всех  $s[i]$ .
5. (27 баллов)  $n \leq 50\,000$
6. (11 баллов) Нет дополнительных ограничений.

## Пример грейдера

Пример грейдера читает входные данные в следующем формате:

- строка 1:  $n \ q$

- строка 2:  $s[0] \ s[1] \ \dots \ s[n-1]$
- строка 3:  $p[0] \ p[1] \ \dots \ p[n-1]$
- строка 4:  $w[0] \ w[1] \ \dots \ w[n-1]$
- строка 5:  $l[0] \ l[1] \ \dots \ l[n-1]$
- строка  $6 + i$  ( $0 \leq i \leq q-1$ ):  $x \ z$  для  $i$ -го вызова функции `simulate`.

Пример грейдера выводит ответы в следующем формате:

- строка  $1 + i$  ( $0 \leq i \leq q-1$ ): возвращаемое значение  $i$ -го вызова функции `simulate`.