

Dungeons Game

Osvojivši zlatnu EGOI medalju, Ema je odlučila napraviti računalnu igru koja će pomoći njenom prijatelju Krešimiru da i on jednog dana postigne sličan uspjeh.

U igri će biti jedan heroj (Krešo), n zadataka i $n + 1$ tamnica. Zadaci su označeni brojevima od 0 do $n - 1$, a tamnice brojevima od 0 do n . Zadatak i ($0 \leq i \leq n - 1$) krvlju je ispisan na zidu tamnice i te ima težinu $s[i]$. U tamnici n ne nalazi se zadatak.

Krešo svoj put do zlata započinje u tamnici x , sa sposobnošću rješavanja zadataka z . Svaki puta kada Krešo uđe u neku tamnicu i ($0 \leq i \leq n - 1$), napast će zadatak i te će se dogoditi jedno od sljedećeg:

- Ako je Krešina sposobnost veća ili jednaka težini zadatka $s[i]$, Krešo će ga uspješno riješiti. Rješavanje zadatka će u tom slučaju rezultirati **porastom** Krešine sposobnosti za $s[i]$ ($s[i] \geq 1$). Nakon toga će se otvoriti portal koji će Krešu odvesti u tamnicu $w[i]$ ($w[i] > i$).
- Inače, zadatak će biti pretežak (odnosno, Krešo prenesposoban) i neće ga uspjeti riješiti. Zbog toga će njegova sposobnost **narasti** za $p[i]$ ($p[i] \geq 1$) (što te ne ubije, to te ojača!). Nakon toga će se otvoriti portal koji će Krešu odvesti u tamnicu $l[i]$.

Primijetite da $p[i]$ može biti manji, jednak ili veći od $s[i]$. Također, $l[i]$ može biti manji, jednak ili veći od i . Neovisno o ishodu "borbe" u tamnici, zadatak ostaje napisan na zidu i nastavlja imati težinu $s[i]$.

Krešo je spreman za zlato kada dođe do tamnice n i tada igra završava. Baš kao i u pravom životu, moguće je dokazati da ste u igri spremni za zlato nakon konačnog broja susreta sa zadacima, neovisno o početnoj sposobnosti i prvom zadatku s kojim ćete se susresti.

Ema je ostavila Krešu da se sam igra, a ona se nastavila baviti zanimljivijim stvarima. Proučavat će odnos početne sposobnosti, prvog zadatka na kojeg natjecatelj naiđe i sposobnosti natjecatelja nakon što je spreman za osvajanje zlata. Najprije će napraviti q simulacija, a u svakoj će simulaciji definirati početnu tamnicu x i sposobnost natjecatelja z . Pomozite joj odrediti, za svaku simulaciju, sposobnost rješavanja zadataka koju će natjecatelj imati nakon što igra završi.

Implementacijski detalji

Potrebno je implementirati sljedeće procedure:

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

- n : broj zadataka.

- s , p , w , l : polja duljine n . Za svaki $0 \leq i \leq n - 1$:
 - $s[i]$ je težina zadatka i . To je također vrijednost za koju će narasti sposobnost natjecatelja ako uspješno riješi i -ti zadatak.
 - $p[i]$ je vrijednost za koju će narasti sposobnost natjecatelja ako ne uspije riješiti i -ti zadatak.
 - $w[i]$ je tamnica u koju će natjecatelj ući nakon što uspješno riješi i -ti zadatak.
 - $l[i]$ je tamnica u koju će natjecatelj ući ako ne uspije riješiti i -ti zadatak.
- Ova procedura će biti pozvana točno jednom, prije prvog poziva procedure `simulate` (vidi dolje).

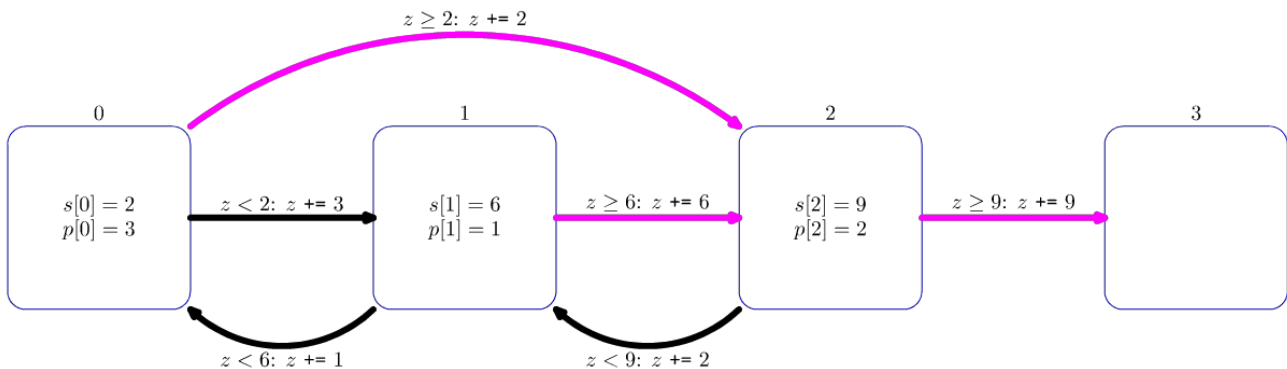
```
int64 simulate(int x, int z)
```

- x : tamnica u kojoj se natjecatelj nalazi na početku.
- z : natjecateljeva početna sposobnost rješavanja zadataka.
- Procedura treba vratiti sposobnost natjecatelja u trenutku kad igra završava, uz pretpostavku da je natjecatelj igru započeo ulaskom u tamnicu x sa sposobnošću z .
- Ova će procedura biti pozvana točno q puta.

Primjer

Promotrimo sljedeći poziv:

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



Gornji dijagram ilustrira ovaj poziv. Svaki kvadrat predstavlja tamnicu. U tamnicama 0, 1 i 2, vrijednosti $s[i]$ i $p[i]$ prikazane su unutar kvadrata. Magenta strelicama prikazane su kretnje natjecatelja u slučaju da riješi odgovarajući zadatak, dok su crnim strelicama prikazane kretnje natjecatelja u slučaju da ne riješi odgovarajući zadatak.

Pretpostavimo da je ocjenjivač pozvao `simulate(0, 1)`.

Igra se odvija na sljedeći način:

Tamnica	Natjecateljeva sposobnost prije rješavanja zadatka	Ishod
0	1	Zadatak nije riješen
1	4	Zadatak nije riješen
0	5	Zadatak je riješen
2	7	Zadatak nije riješen
1	9	Zadatak je riješen
2	15	Zadatak je riješen
3	24	Kraj igre

Stoga, procedura treba vratiti 24.

Pretpostavimo da je ocjenjivač pozvao `simulate(2, 3)`.

Igra se odvija na sljedeći način:

Tamnica	Natjecateljeva sposobnost prije rješavanja zadatka	Ishod
2	3	Zadatak nije riješen
1	5	Zadatak nije riješen
0	6	Zadatak je riješen
2	8	Zadatak nije riješen
1	10	Zadatak je riješen
2	16	Zadatak je riješen
3	25	Kraj igre

Stoga, procedura treba vratiti 25.

Ograničenja

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

Podzadaci

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

2. (26 bodova) $s[i] = p[i]$ (za svaki $0 \leq i \leq n - 1$)
3. (13 bodova) $n \leq 50\,000$, svi su zadaci jednako teški, odnosno, $s[i] = s[j]$ za sve $0 \leq i, j \leq n - 1$.
4. (12 bodova) $n \leq 50\,000$, postoji najviše 5 jedinstvenih vrijednosti među svim vrijednostima $s[i]$.
5. (27 bodova) $n \leq 50\,000$
6. (11 bodova) Nema dodatnih ograničenja.

Ogledni ocjenjivač

Ogledni ocjenjivač čita ulaz u sljedećem obliku:

- redak 1: $n \ q$
- redak 2: $s[0] \ s[1] \ \dots \ s[n - 1]$
- redak 3: $p[0] \ p[1] \ \dots \ p[n - 1]$
- redak 4: $w[0] \ w[1] \ \dots \ w[n - 1]$
- redak 5: $l[0] \ l[1] \ \dots \ l[n - 1]$
- redak $6 + i$ ($0 \leq i \leq q - 1$): $x \ z$ za i -ti poziv procedure `simulate`.

Ogledni ocjenjivač ispisuje vaše odgovore u sljedećem obliku:

- redak $1 + i$ ($0 \leq i \leq q - 1$): izlazna vrijednost i -tog poziva procedure `simulate`.