

Turnir viteških borbi

Za svoje vjenčanje sa Beatrice d'Este 1491. godine, Vojvoda od Milana Lodovico Sforza tražio je od Leonarda da organizuje proslavu vjenčanja koja uključuje veliki viteški turnir koji je trajao cijela tri dana. Međutim, najpopulariji vitez kasni...

Turnir

Na turniru viteških borbi, N vitezova se poreda u jednu vrstu i njihove pozicije u ovoj vrsti se numerišu redom od 0 do $N - 1$. Sudija turnira započinje jednu *rundu* tako što najavi dvije pozicije S i E (gdje je $0 \leq S < E \leq N - 1$). Takmiče se svi vitezovi čije su pozicije između S i E (uključivo): pobjednik nastavlja turnir i vraća se nazad na svoje mjesto u vrsti, s druge strane, svi gubitnici ispadaju iz turnira i napuštaju mjesto takmičenja. Nakon toga, svi preostali vitezovi se skupe tako što se pomijeraju prema prvom mjestu u vrsti na način da zadrže svoj međusobni položaj, stoga su njihove nove pozicije od 0 do $N - (E - S) - 1$. Sudija nakon toga započinje sljedeću rundu, ponavljajući ovaj postupak sve dok ne ostane samo jedan vitez.

Leonardo zna da su svi vitezovi različito snažni. Vitezovi su rangirani po snazi koja je predstavljena cijelim brojem od 0 (najslabiji) do $N - 1$ (najjači). Osim toga zna koje će naredbe izdavati sudija turnira u C rundi: on Leonardno na kraju krajeva... i siguran je da će u svakoj rundi pobijediti najjači vitez.

Vitez koji kasni

$N - 1$ vitezova već je raspoređeno u vrstu, jedino nedostaje najpopularniji vitez. Taj vitez ima rang po snazi R i dolazi nešto kasnije. Da bi turnir učinio zanimljivijim, Leonardo želi da iskoristi popularnost ovog viteza i izabere za njega poziciju u vrsti tako da maksimizira broj rundi u kojima će vitez koji kasni pobijediti. Primijetite da nisu od interesa runde u kojima vitez koji kasni ne učestvuje, nego samo runde u kojima učestvuje i pobjeđuje.

Primjer

Za $N = 5$, $N - 1$ vitezova već je u vrsti i imaju redom rangove po snazi $[1, 0, 2, 4]$. Jasno je da vitez koji kasni ima rang $R = 3$. U $C = 3$ rundi, sudija će najaviti (S, E) pozicije u svakoj rundi u sljedećem redoslijedu: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Ako Leonardo umetne viteza koji kasni na prvu poziciju, rangovi snage cijele vrste vitezova će biti $[3, 1, 0, 2, 4]$. Prva runda uključuje vitezove (na pozicijama 1, 2, 3) sa rangovima 1, 0, 2, pa je pobjednik vitez sa rangom 2. Rangovi snage nove linije su $[3, 2, 4]$. Sljedeća runda je 3 protiv 2 (na pozicijama 0 i 1), te pobjeđuje vitez sa rangom $R = 3$, što ostavlja vrstu $[3, 4]$. U posljednjoj rundi (vitezovi na pozicijama 0, 1) pobjeđuje vitez 4. Prema tome, vitez koji kasni će pobijediti u samo

jednoj rundi (drugoj).

Umjesto toga, ako Leonardo umetne viteza koji kasni između vitezova sa rangovima 1 i 0, vrsta će izgledati ovako: [1, 3, 0, 2, 4]. Sada, prva runda uključuje 3, 0, 2, i vitez sa rangom $R = 3$ pobjeđuje. Sljedeća početna vrsta je [1, 3, 4] i u sljedećoj rundi (1 protiv 3), vitez sa rangom $R = 3$ ponovo pobjeđuje. Krajnja vrsta je [3, 4] gdje 4 pobjeđuje. Prema tome, vitez koji kasni je pobijedio u dvije runde: ovo je ustvari najbolja moguća pozicija obzirom da vitez koji kasni ne može nikako pobijediti u više od dvije runde.

Postavka

Vaš zadatak je da napišete program koji bira najbolju poziciju za viteza koji kasni tako da je broj rundi u kojima on pobjeđuje najveći, kao što Leonardo želi. Konkretno, trebate implementirati funkciju `GetBestPosition(N, C, R, K, S, E)`, gdje su:

- N je broj vitezova;
- C je broj rundi koje najavljuje sudija ($1 \leq C \leq N - 1$);
- R je rang viteza koji kasni; rangovi svih vitezova (kako onih koji su već u vrsti tako i onog koji kasni) su međusobno različiti i iz skupa $0, \dots, N - 1$, a rang R viteza koji kasni dat je eksplicitno iako se može i zaključiti;
- K je niz $N - 1$ cijelih brojeva koji predstavljaju rangove $N - 1$ viteza koji se već nalaze u vrsti;
- S i E su dva niza dužine C : za svako i između 0 i $C - 1$, uključivo, $(i + 1)$ -va runda koju najavljuje sudija će uključivati sve vitezove od pozicije $S[i]$ do pozicije $E[i]$, uključivo. Možete pretpostaviti da će za svako i vrijediti $S[i] < E[i]$.

Svi pozivi ove funkcije su ispravni: $E[i]$ je manje od trenutnog broja vitezova koji su ostali za $(i+1)$ -vu rundu, te će nakon C rundi biti tačno jedan preostali vitez.

`GetBestPosition(N, C, R, K, S, E)` treba da vrati najbolju poziciju P gdje bi Leonardo trebao da ubaci viteza koji kasni ($0 \leq P \leq N - 1$). Ako postoji više takvih pozicija, *vratite najmanju*. (Broj P predstavlja željenu poziciju viteza koji kasni u vrsti takav da je broj 0 prva pozicija. Drugim riječima, P je broj drugih vitezova koji stoje u vrsti ispred viteza u optimalnom rješenju. Specifično, $P = 0$ znači da je vitez koji kasni na početku vrste, dok $P = N - 1$ znači da je na kraju vrste.)

Podzadatak 1 [17 bodova]

Možete smatrati da je $N \leq 500$.

Podzadatak 2 [32 boda]

Možete smatrati da je $N \leq 5\,000$.

Podzadatak 3 [51 bod]

Možete smatrati da je $N \leq 100\,000$.

Implementacijski detalji

Trebate poslati tačno jednu datoteku, pod nazivom `tournament.c`, `tournament.cpp` ili `tournament.pas`. Ova datoteka treba da implementira opisani potprogram sa sljedećim deklaracijama.

C/C++ programi

```
int GetBestPosition(int N, int C, int R, int *K, int *S, int *E);
```

Pascal programi

```
function GetBestPosition(N, C, R : LongInt; var K, S, E : array of LongInt) : LongInt;
```

Ove funkcije trebaju da se ponašaju u skladu sa ranije datim opisima. Naravno, možete implementirati i druge pomoćne funkcije. Vaš program ne smije ni na koji način koristiti standardni ulaz i izlaz niti bilo koju drugu datoteku.

Probni tester

Probni tester dostavljen sa takmičarskim okruženjem očekuje ulazne podatke u sljedećem formatu:

- linija 1: N, C, R ;
- linije 2, ..., N : $K[i]$;
- linije $N + 1$, ..., $N + C$: $S[i], E[i]$.

Vremensko i memorijsko ograničenje

- Vremensko ograničenje: 1 sekunda.
- Memorijsko ograničenje: 256 MiB.