

Riddertoernooi

Bij gelegenheid van zijn bruiloft met Beatrice d'Este in 1491, heeft de hertog van Milaan, Lodovico Sforza, Leonardo gevraagd om de feestelijkheden rond de bruiloft te regisseren, inclusief een groot riddertournooi dat drie hele dagen zal gaan duren. Maar de meest populaire ridder is nog niet gearriveerd ...

Toernooi

In zo'n toernooi worden de N ridders eerst op een rij gezet. Vervolgens worden hun posities genummerd van 0 tot en met $N-1$. De toernooileider laat een ronde beginnen door twee posities af te roepen, S en E (met $0 \leq S < E \leq N - 1$). De ridders met posities vanaf S tot en met E strijden met elkaar; de winnaar neemt zijn positie weer in en de verliezers zijn uitgeschakeld en verlaten het veld. Daarna schuiven de ridders op, in de richting van het begin van de rij, waarbij ze hun onderlinge plek behouden zodat de posities nu genummerd zijn van 0 tot en met $N - (E - S) - 1$. De toernooileider begint dan een nieuwe ronde, en dit proces wordt herhaald tot er slechts één ridder over is.

Leonardo weet dat alle ridders een andere sterkte hebben, aangegeven met een sterkte van 0 (de zwakste) tot en met $N-1$ (de sterkste). Hij weet ook de precieze opdrachten die de toernooileider zal geven voor de C rondes. Leonardo weet zeker dat in iedere ronde de ridder met de hoogste sterkte zal winnen.

Verlate ridder

$N-1$ van de N ridders staan al in de rij. Alleen de meest populaire ridder ontbreekt nog. Deze ridder heeft sterkte R en hij komt wat later. Om het publiek te amuseren wil Leonardo de populariteit van deze ridder gebruiken en hem zo in de rij plaatsen dat hij zo veel mogelijk rondes zal winnen. Let op: we zijn niet geïnteresseerd in de rondes dat deze ridder niet meedoet, enkel in de rondes waarin hij mee doet en wint.

Voorbeeld

Voor $N = 5$ ridders, hebben de $N - 1$ ridders die al in de rij staan de sterktes $[1, 0, 2, 4]$. Hieruit volgt al dat de verlate ridder sterkte $R = 3$ heeft. Voor de $C = 3$ rondes, is de toernooileider van plan de volgende (S, E) posities af te roepen, in deze volgorde: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Als Leonardo de verlate ridder vooraan zet, dan zijn de sterktes van de de ridders in de rij achtereenvolgens $[3, 1, 0, 2, 4]$. De eerste ronde betreft de ridders (op de posities 1, 2 en 3) met de sterktes 1, 0 en 2, met de ridder met sterkte 2 als winnaar. De nieuwe rij is $[3, 2, 4]$. In de volgende ronde spelen 3 tegen 2 (posities 0 en 1) en de ridder met sterkte $R=3$ wint. De rij is nu $[3, 4]$. De

laatste ronde (met de posities 0 en 1) heeft 4 als winnaar. De verlate ridder wint dus maar één ronde (de tweede).

Als Leonardo daarentegen de verlate ridder plaatst tussen de ridders met de sterktes 1 en 0, ziet de rij er als volgt uit: [1, 3, 0, 2, 4]. De eerste ronde gaat nu tussen 3, 0 en 2 en de ridder met $R=3$ wint. De volgende rij is [1, 3, 4] en in die ronde (1 tegen 3) wint de ridder met $R=3$ opnieuw. De laatste ronde is met rij [3, 4], waarbij 4 wint. De verlate ridder wint twee rondes: dit is inderdaad de best mogelijke plek om hem in de rij te zetten omdat de verlate ridder op geen enkele manier meer dan twee rondes kan winnen.

Opdracht

Schrijf een programma dat de beste positie voor de verlate ridder in de rij vindt, zodat het aantal rondes dat hij wint maximaal is. Je moet een routine implementeren die `GetBestPosition(N, C, R, K, S, E)` heet, waarvoor geldt:

- N is het aantal ridders;
- C is het aantal rondes dat de toernooileider zal afroepen ($1 \leq C \leq N - 1$);
- R is de sterkte van de verlate ridder; de sterktes van alle ridders (zowel die in de rij als de verlate ridder) zijn allemaal anders en ze lopen van 0 tot en met $N - 1$, en sterkte R van de verlate wordt apart opgegeven, zelfs al is het af te leiden;
- K is een array van $N - 1$ integers, de sterktes van de $N - 1$ ridders die al in de rij staan;
- S en E zijn twee arrays met lengte C : voor elke i van 0 tot en met $C - 1$ geldt, tijdens de $(i + 1)$ e ronde van het toernooi zijn vechten de ridders van positie $S[i]$ tot en met positie $E[i]$. Je kunt ervan uitgaan dat voor elke i geldt $S[i] < E[i]$.

Alle aanroepen van deze routine zijn geldig: $E[i]$ is altijd lager dan het aantal ridders dat overgebleven is voor de $(i+1)$ e ronde, en na de C aanroepen zal er precies één ridder over zijn.

`GetBestPosition(N, C, R, K, S, E)` moet de beste positie P teruggeven waarop Leonardo de verlate ridder moet plaatsen ($0 \leq P \leq N - 1$). Als er meer gelijkwaardige posities zijn, voer dan het laagste nummer uit. (De positie P is de positie van de verlate ridder in de uiteindelijke rij, waarbij de rij begint bij positie 0. Met andere woorden: er staan precies P ridders voor de verlate ridder in de rij in de optimale oplossing. In het bijzonder geldt $P = 0$ als de ridder vooraan de rij komt en $P = N - 1$ betekent dat hij helemaal achteraan wordt geplaatst.)

Subtask 1 [17 punten]

Je mag er van uit gaan dat $N \leq 500$.

Subtask 2 [32 punten]

Je mag er van uitgaan dat $N \leq 5\,000$.

Subtask 3 [51 punten]

Je mag er van uitgaan dat $N \leq 100\,000$.

Implementatie details

Je moet precies één programmabestand inzenden, met de naam `tournament.c`, `tournament.cpp` of `tournament.pas`. Dit programma moet de hierboven omschreven routine implementeren met de volgende kenmerken.

C/C++ programma's

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

Pascal programma's

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

Deze routine moet zich gedragen zoals hierboven beschreven. Natuurlijk staat het je vrij andere deelprogramma's te schrijven voor intern gebruik. Je inzending mag geen gebruik maken van standard input of standard output, of van andere bestanden.

Voorbeeld graders

De voorbeeld grader die bij de wedstrijdgeving hoort verwacht de invoer in het volgende format:

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

Tijds- en geheugenlimieten

- Tijdslimit: 1 seconde.
- Geheugenlimiet: 256 MiB.