

Turnir viteških bojev

Za svojo poroko z Beatrice d'Este leta 1491 je vojvoda Milana, Lodovico Sforza, prosil Leonarda, naj orkestrira svatbo, ki vključuje velik viteški turnir, kateri traja cele tri dni. A kot ponavadi, vitez Matteo zamuja...

Turnir

Na turnirju viteških bojev se N vitezov razvrsti v eno vrsto in njihove pozicije so oštevilčene po vrsti, od 0 do $N - 1$. Sodnik turnirja prične *krog* tako, da razglasi dve poziciji S in E (kjer je $0 \leq S < E \leq N - 1$). V tem krogu tekmujejo vitezi, katerih pozicije so med S in E (vključujoče). Zmagovalec nadaljuje turnir, zato se vrne na svoje začetno mesto v vrsti, vsi poraženci pa izpadejo iz turnirja in zapustijo bojišče. Preostali vitezi se za tem prestavijo proti začetku vrste tako, da ohranijo začetno medsebojno ureditev. Njihove nove pozicije so tako od 0 do $N - (E - S) - 1$. Sodnik nato razglasi naslednji krog ter nadaljuje opisani postopek, vse dokler ne preostane le en vitez.

Leonardo ve, da so vsi vitezi različno močni. Vitezi so rangirani po moči, od 0 (najslabši) do $N - 1$ (najmočnejši). Poleg tega pozna tudi natančen vrstni red sodnikovih razglasitev vseh C krogov — konec koncev, gre za Leonarda... prepričan je tudi, da bo vsakem od teh krogov zmagal najmočnejši vitez.

Vitez, ki zamuja

$N - 1$ vitezov je že postavljenih v vrsto, manjka le še najbolj priljubljen vitez Matteo. On ima rang moči R , prikazal pa se bo z nekoliko zamude. Leonardo želi izkoristiti njegovo priljubljenost za zanimivejši turnir ter mu izbrati takšno mesto v vrsti, da bo število krogov, v katerih bo Matteo zmagal, čim večje. Bodi pozoren na to, da nas ne zanimajo krogi, pri katerih Matteo ne sodeluje, temveč le krogi, v katerih prisostvuje in v njih zmaga.

Primer

Za $N = 5$ je rangiranje vitezov, ki so že v vrsti $[1, 0, 2, 4]$; Matteo ima rang $R = 3$. V $C = 3$ krogih bo sodnik razglasil (S, E) pozicije posameznih krogov v naslednjem vrstnem redu: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Če Leonardo postavi Mattea na prvo mesto, bo rangiranje vitezov v vrsti $[3, 1, 0, 2, 4]$. Prvi krog vključuje viteze (s pozicij 1, 2, 3) z rangi 1, 0, 2. To privede do zmage viteza z rangom 2 in nove vrste, ki je $[3, 2, 4]$. V naslednjem krogu se pomerita vitez z rangi 3 in 2 (s pozicij 0, 1), zmaga vitez z rangom 3 in vrsta se preoblikuje v $[3, 4]$. Zadnji krog (s pozicij 0, 1) ima 4 za zmagovalca. Matteo je v tem primeru zmagal le v enem krogu turnirja (drugega).

Leonardo bi Mattea lahko vstavil med viteza z rangi 1 in 0, tako da bi vrsta izgledala takole: $[1, 3,$

0, 2, 4]. Tokrat prvi krog vključuje 3, 0, 2, in Matteo zmaga. Naslednja razporeditev je [1, 3, 4] in Matteo ponovno zmaga. Zadnja vrsta je [3, 4], kjer zmaga 4. Matteo je zmagal dva kroga in to je v bistvu najboljša možna umestitev, saj razporeditev, kjer bi Matteo zmagal več kot dvakrat, ne obstaja.

Naloga

Tvoja naloga je napisati program, ki bo našel najboljšo možno umestitev za Mattea. Leonardo si želi razporeditev, ki maksimizira število krogov v katerih Matteo zmaga.

- N je število vitezov;
- C je število krogov katere sodnik razglasi ($1 \leq C \leq N - 1$);
- R je rang Mattea — viteza, ki vedno zamudi. Rangji vseh vitezov (tako tistih, ki so že v vrsti, kot tega, ki zamuja) so permutacija števil $0, \dots, N - 1$. Rang poznega viteza R je podan posebej, kljub temu da ga je moč izpeljati;
- K je seznam $N - 1$ celih števil in predstavlja rangiranje $N - 1$ vitezov, ki že stojijo v začetni vrsti;
- S in E sta dva seznama velikosti C : za vsak i med 0 in $C - 1$ (vključujoče) bo $(i + 1)$ -ti krog, katerega najavi sodnik, vključeval vse viteze s pozicij od $S[i]$ do $E[i]$ (vključujoče). Lahko privzameš, da za vsak i velja $S[i] < E[i]$.

Vsi klici te funkcije so pravilni: $E[i]$ je manjši od trenutnega števila vitezov ki se prebijejo do $(i + 1)$ -tega kroga, ter da bo po zaključenem C -tem krogu ostal natanko en vitez.

`GetBestPosition(N, C, R, K, S, E)` mora vrniti najboljšo pozicijo P , kamor mora Leonardo umestiti viteza ki vedno zamuja ($0 \leq P \leq N - 1$). Če obstaja več enakovrednih umestitev, vrni *najmanjšo*. (Umestitvena pozicija P se začne z 0 in predstavlja mesto v končni vrsti, kjer bo Matteo stal. Z drugimi besedami, P je število vitezov, ki v optimalni rešitvi stojijo pred Matteom. Specifično, $P = 0$ pomeni, da bo Matteo stal na samem začetku vrste in $P = N - 1$ pomeni, da bo stal na samem koncu vrste.)

1. podnaloga [17 točk]

Lahko privzameš, da je $N \leq 500$.

2. podnaloga [32 točk]

Lahko privzameš, da je $N \leq 5\,000$.

3. podnaloga [51 točk]

Lahko privzameš, da je $N \leq 100\,000$.

Podrobnosti implementacije

Oddati moraš natanko eno datoteko, poimenovano `tournament.c`, `tournament.cpp` ali `tournament.pas`. Ta datoteka mora vsebovati implementacijo zgoraj opisanega podprograma in uporabljati sledeče podpise.

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;
```

Ti podprogrami se morajo vesti, kot je opisano zgoraj. Seveda, *poetica di programmazione*, imaš vso svobodo implementirati svoje podprograme za lastno uporabo. Tvoje oddaje ne smejo nikakor uporabljati standardnega vhoda/izhoda, niti katerihkoli drugih datotek.

Primer ocenjevalca

Nalogi priložen ocenjevalec bo pričakoval vhodne podatke v naslednji obliki:

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

Omejitve časa in porabe spomina

- Omejitev časa: 1 sekunda.
- Omejitev spomina: 256 MiB.