

Jousting tournament

Untuk pernikahannya dengan Beatrice d'Este pada tahun 1491, Duke of Milan Lodovico Sforza meminta Leonardo merancang pesta pernikahannya, termasuk sebuah "jousting tournament" (pertandingan berkelahi dengan tombak sambil berkuda) yang berlangsung selama tiga hari penuh. Tapi ksatria yang paling populer terlambat datang...

Pertandingan

Pada sebuah jousting tournament, N ksatria diatur dalam sebuah baris dan posisinya dinomori dari 0 hingga $N - 1$ sesuai dengan urutannya pada baris. Joust master mengatur sebuah *putaran* dengan memanggil dua posisi S dan E (di mana $0 \leq S < E \leq N - 1$). Semua ksatria yang posisinya antara S dan E (inklusif) bertanding: pemenangnya melanjutkan pertandingan dan kembali ke tempatnya pada baris, sedangkan yang kalah keluar dari permainan dan meninggalkan lapangan. Setelah itu, ksatria yang tersisa dirapatkan ke arah awal baris, dengan tetap mempertahankan urutan relatif mereka pada baris, sehingga hasil posisi mereka adalah dari 0 hingga $N - (E - S) - 1$. Joust master mengadakan putaran berikutnya, dan mengulang permainan sampai tersisa hanya 1 ksatria.

Leonardo tahu bahwa semua ksatria mempunyai kekuatan yang berbeda-beda yang direpresentasi sebagai nilai ranks berbeda dari 0 (paling lemah) hingga $N - 1$ (paling kuat). Dia juga mengetahui perintah-perintah yang akan dipanggil oleh joust master pada C putaran: tidak heran, karena dia adalah Leonardo... dan dia yakin bahwa pada masing-masing putaran ini ksatria dengan rank tertinggi akan menang.

Ksatria Terlambat

Sebanyak $N - 1$ dari N ksatria sudah diatur pada baris, hanya ksatria yang paling populer yang tertinggal. Ksatria ini mempunyai rank R dan datang sedikit terlambat. Demi menjaga agar pertandingan tetap seru, Leonardo ingin memanfaatkan popularitasnya dan memilihkan untuknya sebuah posisi pada baris yang akan memaksimalkan banyaknya putaran permainan yang pada akhirnya akan dimenangkan oleh sang ksatria yang terlambat. Perhatikan bahwa kita tidak tertarik pada putaran yang tidak melibatkan ksatria terlambat, melainkan hanya putaran di mana ia berpartisipasi dan menang.

Contoh

Untuk $N = 5$ ksatria, $N - 1$ ksatria yang sudah diatur dalam baris mempunyai ranks $[1, 0, 2, 4]$. Konsekuensinya, ksatria terlambat mempunyai rank $R = 3$. Untuk $C = 3$ putaran, sang joust master bermaksud memanggil posisi (S, E) per putaran, dengan urutan sebagai berikut: $(1, 3), (0, 1), (0, 1)$.

Jika Leonardo menyisipkan ksatria yang terlambat pada posisi pertama, ranks dari para ksatria pada

baris akan menjadi [3, 1, 0, 2, 4]. Putaran pertama melibatkan ksatria-ksatria (pada posisi 1, 2, 3) dengan ranks 1, 0, 2, menyisakan ksatria dengan rank 2 sebagai pemenang. Baris yang baru adalah [3, 2, 4]. Putaran berikutnya adalah 3 melawan 2 (pada posisi 0, 1), dan ksatria dengan rank $R = 3$ menang, menyisakan baris menjadi [3, 4]. Putaran terakhir (pada posisi 0, 1) menghasilkan 4 sebagai pemenang. Maka sang ksatria terlambat hanya menang satu putaran (yang kedua).

Malahan, jika Leonardo menyisipkan ksatria yang terlambat di antara dua ksatria yang mempunyai rank 1 dan 0, barisnya akan terlihat seperti ini: [1, 3, 0, 2, 4]. Kali ini, putaran pertama melibatkan 3, 0, 2, dan ksatria dengan rank $R = 3$ menang. Baris berikutnya adalah [1, 3, 4], dan pada putaran berikutnya (1 melawan 3) ksatria dengan rank $R = 3$ akan menang lagi. Baris terakhir adalah [3, 4], di mana 4 menang. Maka, ksatria yang terlambat menang dua putaran: ini merupakan penempatan terbaik karena tidak ada cara lain dimana ksatria yang terlambat akan menang lebih dari dua kali.

Statement

Tugas anda adalah menulis sebuah program yang akan memilih posisi terbaik untuk ksatria yang terlambat sehingga banyak putaran yang dimenangkannya akan maksimal seperti yang dikehendaki oleh Leonardo. Khususnya, Anda harus mengimplementasi sebuah routine yang dinamakan `GetBestPosition(N, C, R, K, S, E)`, di mana:

- N adalah banyaknya ksatria;
- C adalah banyaknya putaran yang dipanggil oleh joust master ($1 \leq C \leq N - 1$);
- R adalah rank dari ksatria yang terlambat; ranks dari semua ksatria (semua yang sudah diberiskan dan ksatria yang terlambat) berbeda dan dipilih dari 0, ..., $N - 1$ dan rank R dari ksatria yang terlambat diberikan secara eksplisit walaupun sebetulnya dapat disimpulkan.
- K adalah sebuah array terdiri dari $N - 1$ bilangan bulat yang merepresentasikan ranks dari $N - 1$ ksatria yang ada di baris pada keadaan awal;
- S dan E adalah array dengan ukuran C : untuk setiap i antara 0 dan $C - 1$, inklusif, putaran ke- $(i + 1)$ yang dipanggil oleh joust master akan menyertakan semua ksatria yang berada pada posisi $S[i]$ hingga posisi $E[i]$, inklusif. Anda boleh berasumsi bahwa untuk setiap i , berlaku $S[i] < E[i]$.

Pemanggilan ke routine pasti valid: $E[i]$ lebih kecil dari banyaknya ksatria yang tersisa pada putaran ke- $(i + 1)$, dan setelah semua perintah C hanya ada tepat satu ksatria yang tersisa.

`GetBestPosition(N, C, R, K, S, E)` harus mengembalikan posisi terbaik P di mana Leonardo harus meletakkan ksatria yang terlambat ($0 \leq P \leq N - 1$). Jika ada lebih dari satu posisi ekuivalen yang mungkin, *keluarkan yang paling kecil*. (Posisi P adalah posisi 0-based dari ksatria yang terlambat pada baris. Dengan kata lain, P adalah banyaknya ksatria lain yang berdiri sebelum ksatria yang terlambat pada solusi optimal. Khususnya, $P = 0$ berarti ksatria yang terlambat berada pada awal baris, dan $P = N - 1$ berarti ia berada pada posisi terakhir.)

Subtask 1 [17 points]

Anda boleh berasumsi bahwa $N \leq 500$.

Subtask 2 [32 points]

Anda boleh berasumsi bahwa $N \leq 5\,000$.

Subtask 3 [51 points]

Anda boleh berasumsi bahwa $N \leq 100\,000$.

Detail Implementasi

Anda harus mengumpulkan tepat sebuah file yang diberi nama `tournament.c`, `tournament.cpp` atau `tournament.pas`. File ini harus mengimplementasikan subprogram yang dideskripsikan di atas dengan signature sebagai berikut.

Program C/C++

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

Program Pascal

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

Subprogram-subprogram ini harus berperilaku seperti yang dijelaskan di atas. Tentu saja Anda boleh mengimplementasi subprogram lain untuk keperluan internal mereka. Program yang Anda kumpulkan tidak boleh berinteraksi dengan cara apapun melalui standard input/output, maupun melalui file lain.

Contoh grader

Contoh grader yang disediakan pada soal mengharapkan input dengan format sebagai berikut:

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

Batas Waktu dan Memori

- Batas waktu: 1 detik.
- Batas memori: 256 MiB.