

Torneo de Justas

Para su boda con Beatrice d'Este in 1.491, el Duque de Milán Lodovico Sforza le pidió a Leonardo dirigir las celebraciones de la boda, incluyendo un gran torneo de justas que duró tres días completos. Pero el caballero más popular está atrasado....

El Torneo

En un torneo de justas, los N caballeros están primero formados en una fila y luego sus posiciones son numeradas de 0 a $N - 1$ siguiendo el orden de la fila. El jefe de las justas define una *ronda* eligiendo dos posiciones S y E (donde $0 \leq S < E \leq N - 1$). Todos los caballeros cuyas posiciones están entre S y E (inclusive) compiten: el ganador continúa en el torneo y regresa a su puesto en la fila, en cambio los perdedores están fuera del juego y dejan el campo. Después de eso, los caballeros restantes se agrupan al comienzo de la fila, preservando su orden relativo en la fila, de tal manera que sus posiciones están ahora numeradas de 0 a $N - (E - S) - 1$. El jefe de justas llama otra ronda, repitiendo este proceso hasta que quede un caballero.

Leonardo sabe que todos los caballeros tienen fortalezas diferentes representadas como rangos distintos desde 0 (más débil) hasta $N - 1$ (más fuerte). El también sabe los comandos exactos que el jefe de justas dará para las C rondas: después de todo, él es Leonardo... y él está seguro que en cada una de estas rondas el caballero con rango más alto ganará.

Caballero atrasado

$N - 1$ de los caballeros ya están formados en la fila, pero falta el caballero más popular. Este caballero tiene rango R y está llegando un poco tarde. Para beneficio del espectáculo, Leonardo quiere explotar su popularidad y elegir para él una posición en la fila que maximice el número de rondas que ganará el caballero atrasado. Note que no estamos interesados en las rondas que no involucren al caballero atrasado, únicamente estamos interesados en las rondas en las cuales él tome parte y gane.

Ejemplo

Para $N = 5$ caballeros, los $N - 1$ caballeros que ya están agrupados en la fila tienen rangos $[1, 0, 2, 4]$, respectivamente. En consecuencia, el caballero atrasado tiene rango $R = 3$. Para las $C = 3$ rondas, el jefe de justas quiere llamar las posiciones (S, E) por ronda, en este orden: $(1, 3)$, $(0, 1)$, $(0, 1)$.

Si Leonardo inserta al caballero atrasado en la primera posición, los rangos de los caballeros en la fila serán $[3, 1, 0, 2, 4]$. La primera ronda involucra a los caballeros (en las posiciones 1, 2, 3) con rangos 1, 0, 2, dejando al caballero con rango 2 como el ganador. La nueva fila es $[3, 2, 4]$. La

siguiente ronda es 3 contra 2 (en las posiciones 0, 1), y el caballero con rango $R = 3$ gana, dejando la fila [3, 4]. La ronda final (en las posiciones 0, 1) tiene a 4 como ganador. Por lo tanto, el caballero atrasado ganará únicamente una ronda (la segunda).

En lugar de esto, si Leonardo inserta al caballero atrasado entre esos dos de rangos 1 y 0, la fila se ve de esta manera [1, 3, 0, 2, 4]. Esta vez, la primera ronda involucra 3, 0, 2, y el caballero con rango $R = 3$ gana. La siguiente fila es [1, 3, 4] y en la siguiente ronda con (1 contra 3) el caballero con rango $R = 3$ gana nuevamente. La fila final es [3, 4], en la cual gana 4. Por lo tanto, el caballero atrasado gana dos rondas: esta es la mejor ubicación posible pues no hay otra forma en que el caballero atrasado gane más de dos veces.

Problema

Tu tarea es escribir un programa que elija la mejor posición para el caballero atrasado de tal manera que el número de rondas que él gane sea maximizado, como quiere Leonardo. Específicamente, tienes que implementar una rutina llamada `GetBestPosition(N, C, R, K, S, E)`, donde:

- N es el número de caballeros;
- C es el número de rondas llamadas por el jefe de justas ($1 \leq C \leq N - 1$);

R es el rango del caballero atrasado; los rangos de todos los caballeros (tanto de los que ya están en la fila como del atrasado) son distintos y elegidos de $0, \dots, N - 1$ y el rango R del caballero atrasado se da explícitamente aunque puede ser deducido;

- K es un arreglo de $N - 1$ enteros, representando los rangos de los $N - 1$ caballeros que ya están en la fila inicial;
- S y E son dos arreglos de tamaño C : para cada i entre 0 y $C - 1$, inclusive, la ronda $(i + 1)$ ésima llamada por el jefe de justas involucrará a los caballeros desde la posición $S[i]$ a la posición $E[i]$, inclusive. Usted puede asumir que para cada i , $S[i] < E[i]$.

Las llamadas pasadas a esta rutina son válidos: tenemos que $E[i]$ es menor que el número actual de caballeros que quedan en la ronda $(i + 1)$ ésima, y que después de C comandos habrá exactamente un caballero restante.

`GetBestPosition(N, C, R, K, S, E)` debe devolver la mejor posición P donde Leonardo debería poner al caballero atrasado ($0 \leq P \leq N - 1$). Si hay varias posiciones equivalentes, dé la menor. (La posición P es la posición basada en 0 del caballero atrasado en la fila resultante. En otras palabras, P es el número de otros caballeros que están en la fila antes que el caballero en la solución óptima. Específicamente, $P = 0$ quiere decir que el caballero atrasado está al comienzo de la fila y $P = N - 1$ quiere decir que está al final de ella.)

Sub-tarea 1 [17 puntos]

Puedes asumir que $N \leq 500$.

Sub-tarea 2 [32 puntos]

Puedes asumir que $N \leq 5\,000$.

Sub-tarea 3 [51 puntos]

Debes asumir que $N \leq 100\,000$.

Detalles de implementacion

Debes enviar un sólo archivo, llamado `tournament.c`, `tournament.cpp` o `tournament.pas`. Dicho archivo debe implementar una sub-rutina con el siguiente encabezado:

Programas en C/C++

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

Programas en Pascal

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

Estas subrutinas deben comportarse como se ha descrito anteriormente. Por supuesto, eres libre de implementar otras subrutinas para uso interno. Tu envío no debe interactuar de ninguna manera con la entrada/salida estándar, ni con ningún otro archivo.

Ejemplo de Evaluador

El calificador ejemplo proporcionado con el ambiente de la tarea esperará entrada en el siguiente formato:

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

Límites de Tiempo y Memoria

- Tiempo límite: 1 segundo.
- Memoria límite: 256 MiB.