Torneo de justas

Para su boda con Beatrice d'Este en 1491, el Duque de Milán Lodovico Sforza pidió a Leonardo que orquestrara las celebraciones de la boda, incuyendo un gran torneo de justas que duró tres días. Pero el caballero más popular llegó tarde…

Torneo

En un torneo de justas, los N caballeros primero se colocan en fila, y sus posiciones se numeran de 0 a N - 1 siguiendo el orden en la fila. El maestro de justas *llama una ronda* escogiendo dos posiciones S y E (con $0 \leq S < E \leq N - 1$). Compiten todos los caballeros cuyas posiciones estén comprendidas entre S y E (inclusive): el ganador continúa en el torneo y retorna a su posición en la fila, mientras que todos los otros quedan fuera del juego. Después, los caballeros restantes se compactan hacia el principio de la fila, preservando su orden relativo, de manera que las posiciones resultantes están entre 0 y N - (E - S) - 1. El maestro de justas sigue llamando rondas hasta que quede un caballero.

Leonardo sabe que todos los caballeros tienen fuerzas diferentes, representadas por rangos distintos entre 0 (más débil) y N - 1 (más fuerte). También sabe la cantidad exacta de llamadas del maestro de justas durante las C rondas: él es Leonardo, después de todo… y está seguro que en cada una de las rondas ganará el caballero con el rango más alto.

Caballero tardón

N - 1 de los N caballeros ya están en la fila, y sólo falta el caballero más popular. Este caballero tiene rango R y llega tarde. Leonardo quiere explotar su popularidad, así que escoge para él una posición que maximice el número de rondas que ganará. Fijáos que no estamos interesados en las rondas que no involucran al caballero tardón, sólo en las que sí toma parte y gana.

Ejemplo

Para N = 5 caballeros, los N - 1 caballeros ya preparados tienen rangos [1, 0, 2, 4]. Por lo tanto, el caballero tardón tiene rango R = 3. Para las C = 3 rondas, el maestro de justas llamará las posiciones (S, E) en este orden: (1, 3), (0, 1), (0, 1).

Si Leonardo inserta el caballero tardón en la primera posición, los rangos de la fila serán [3, 1, 0, 2, 4]. La primera ronda involucra los caballeros (en las posiciones 1, 2, 3) con rangos 1, 0, 2, dejando como ganador al caballero de rango 2. La nueva fila es [3, 2, 4]. La próxima ronda es 3 contra 2 (en las posiciones 0, 1), y el caballero con rango R = 3 gana, dejando [3, 4]. La última ronda (posiciones 0, 1) tiene a 4 como ganador. Por lo tanto, el caballero tardón solo gana una ronda (la segunda).

En cambio, si Leonardo inserta el caballero tardón entre los de rango 1 y 0, la fila será: [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 (1 contra 3) el caballero con rango R = 3 gana otra vez. La última fila es [3, 4], donde 4 gana. Por lo tanto el caballero tardón gana dos rondas: este es el mejor emplazamiento, ya que no existe ninguna otra situación donde el caballero tardón gane más de dos rondas.

Enunciado

Tu tarea consiste en escribir un programa que escoja la mejor posición para el caballero tardón, de manera que se maximice el número de rondas que ganará. Específicamente, tienes que implementar una rutina con nombre `GetBestPosition`(N, C, R, K, S, E), donde:

- N es el número de caballeros;
- C es el número de rondas ($1 \leq C \leq N - 1$);
- R es el rango del caballero tardón; el rango de todos los caballeros (tanto del tardón como de los que ya estan en fila) son números distintos entre 0, ..., N-1, y el rango R del caballero tardón se da explícitamente, aunque se podría deducir.
- K es un array de N - 1 enteros, que representa los rangos de los N - 1 caballeros que estan en la fila inicial;
- S y E son dos arrays de tamaño C: para cada i entre 0 y C - 1, inclusive, la (i + 1)-ésima ronda incluye todos los caballeros desde la posición S[i] hasta la posición E[i], inclusive. Puedes asumir S[i] < E[i] para cada i.

Las llamadas a esta rutina són válidas: E[i] es menor que el número actual de de caballeros restantes para la (i + 1)-ésima ronda, y después de todas las C llamadas quedará exactamente un caballero.

`GetBestPosition`(N, C, R, K, S, E) tiene que devolver la mejor posición P (empezando en 0) para el caballero tardón ($0 \leq P \leq N - 1$). En otras palabras, P es el número de caballeros situados antes del caballero tardón en la solución óptima. Por ejemplo, con P = 0 el caballero tardón está al inicio de la fila, y con P = N - 1 está al final. Si hay múltiples posiciones equivalentes, devuelve la más pequeña.

Subtarea 1 [17 puntos]

Puedes asumir que $N \leq 500$.

Subtarea 2 [32 puntos]

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

Subtarea 3 [51 puntos]

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

Detalles de implementación

Tienes que enviar exactamente un archivo con nombre `tournament.c`, `tournament.cpp` o `tournament.pas`. Este tiene que implementar la rutina descrita anteriormente usando la siguiente cabecera.

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 rutinas tienen que comportarse como se ha descrito anteriormente. Por supuesto, estas rutinas pueden llamar a otras rutinas que hayas programado. Tu envío no debe interactuar de ningún modo con la entrada/salida estándar, ni con ningún otro fichero.

Sample graders

El sample grader que tienes disponible espera un archivo de entrada con el siguiente formato:

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

Restricciones de tiempo y memoria

- Time limit: 1 second.
- Memory limit: 256 MiB.