



## International Olympiad in Informatics 2015

26th July - 2nd August 2015

Almaty, Kazakhstan

Day 1

teams

Language: en-BOL

# Equipos

Hay una clase de  $N$  estudiantes, enumerados de  $0$  a  $N - 1$ . Cada día el profesor de la clase tiene algunos proyectos para los estudiantes. Cada proyecto tiene que ser completado por un equipo de estudiantes ese mismo día. Los proyectos pueden tener diferentes dificultades. Para cada proyecto, el profesor sabe exactamente el tamaño del equipo que debe trabajar en él.

Diferentes estudiantes pueden preferir diferentes tamaños de equipo. Más precisamente, el estudiante  $i$  solo puede ser asignado a un equipo de tamaño entre  $A[i]$  y  $B[i]$ , inclusivos. Cada día, un estudiante puede ser asignado máximo a un equipo. Algunos estudiantes pueden no ser asignados a ningún equipo. Cada equipo trabajará en un proyecto.

El profesor ha escogido los proyectos para cada uno de los siguientes  $Q$  días. Para cada uno de estos días, determine si es posible asignar estudiantes a los equipos de tal manera que exista un equipo trabajando en cada proyecto.

# Ejemplo

Suponga que hay  $N = 4$  estudiantes y  $Q = 2$  días. Las restricciones de los estudiantes sobre los tamaños de los equipos estan dadas en la tabla a continuación.

Estudiante	0	1	2	3
$A$	1	2	2	2
$B$	2	3	3	4

En el primer día hay  $M = 2$  proyectos. Los tamaños requeridos de los equipos son  $K[0] = 1$  y  $K[1] = 3$ . Estos dos equipos pueden ser formados asignando al estudiante 0 a un equipo de tamaño 1 y los otros tres estudiantes a un equipo de tamaño 3.

En el segundo día hay  $M = 2$  proyectos otra vez, pero esta vez los tamaños requeridos de los equipos son  $K[0] = 1$  y  $K[1] = 1$ . En este caso no es posible formar los equipos, ya que solo hay un estudiante que puede estar en un equipo de tamaño 1.

# Tarea

Dada la descripción de todos los estudiantes:  $N$ ,  $A$ , y  $B$ , así como la secuencia de  $Q$  preguntas — una por cada día. Cada pregunta consiste de el número  $M$  de proyectos en ese día y una secuencia  $K$  de longitud  $M$  conteniendo los tamaños requeridos de los equipos. Para cada pregunta, su programa debe retornar si es posible formar todos los equipos.

Debe implementar las funciones `init` y `can`:

- `init(N, A, B)` — será la primera función que el evaluador (grader) llamará, y la llamará

exactamente una vez.

- $N$ : el número de estudiantes.
- $A$ : arreglo de longitud  $N$ :  $A[i]$  el mínimo tamaño de equipo para el estudiante  $i$ .
- $B$ : arreglo de longitud  $N$ :  $B[i]$  el máximo tamaño de equipo para el estudiante  $i$ .
- La función no tiene valor de retorno.

Puede asumir que  $1 \leq A[i] \leq B[i] \leq N$  para cada  $i = 0, \dots, N-1$ .

- $\text{can}(M, K)$  — Después de llamar `init`, el evaluador llamará esta función  $Q$  veces consecutivas, una vez por cada día.
  - $M$ : el número de proyectos para este día.
  - $K$ : un arreglo de longitud  $M$  conteniendo el tamaño de equipo requerido para cada uno de los proyectos.
  - La función debe retornar 1 si es posible formar todos los equipos requeridos y 0 en otro caso.
  - Puede asumir que  $1 \leq M \leq N$ , y que para cada  $i = 0, \dots, M-1$  se tiene  $1 \leq K[i] \leq N$ . Note que la suma de todos los  $K[i]$  puede exceder  $N$ .

## Subtareas

Denotemos con  $S$  la suma de los valores de  $M$  en todos los llamados a  $\text{can}(M, K)$ .

Subtarea	puntos	$N$	$Q$	Restricciones adicionales
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	Ninguna
2	13	$1 \leq N \leq 100,000$	$Q = 1$	Ninguna
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

## Evaluador de ejemplo

El evaluador ejemplo lee la entrada en el siguiente formato:

- línea 1:  $N$
- líneas 2, ...,  $N+1$ :  $A[i] B[i]$
- línea  $N+2$ :  $Q$
- líneas  $N+3, \dots, N+Q+2$ :  $M K[0] K[1] \dots K[M-1]$

For each question, the sample grader prints the return value of `can`.