



## Teams

Hay un grupo de  $N$  estudiantes, numerados desde  $0$  hasta  $N - 1$ . Cada día el profesor del grupo tiene algunos proyectos para los estudiantes. Cada proyecto tiene que ser completado por un equipo de estudiantes dentro del mismo día. Los proyectos pueden tener variadas dificultades. Para cada proyecto, el profesor sabe el tamaño exacto del equipo que debe trabajar en él.

Diferentes estudiantes pueden preferir diferentes tamaños de equipos. Más exacto, el estudiante  $i$  puede solamente estar asignado a un equipo de tamaño entre  $A[i]$  y  $B[i]$  inclusive. En cada día, un estudiante puede ser asignado a lo sumo a un equipo. Algunos estudiantes pudieran no estar asignados a algunos equipos. Cada equipo trabajará en un proyecto simple.

El profesor ya ha elegido los proyectos para cada uno de los próximos  $Q$  días. Para cada uno de estos días, determine si es posible asignar estudiantes a equipos tal 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 el tamaño de los equipos son dadas en la tabla de abajo.

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

En el primer día hay  $M = 2$  proyectos. Los tamaños de equipo requeridos son  $K[0] = 1$  y  $K[1] = 3$ . Esos dos equipos pueden estar formados por la asignación del estudiante 0 al equipo de tamaño 1 y los tres estudiantes restantes al 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, pues hay solamente un estudiante quien puede estar en un equipo de tamaño 1.

## Tarea

A usted le son dadas la descripción de todos los estudiantes:  $N$ ,  $A$ , y  $B$ , así como una secuencia de  $Q$  preguntas — una acerca de cada día. Cada pregunta consiste de un 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 tiene que retornar si es posible formar todos los equipos.

Usted necesita implementar las funciones `init` y `can`:

- `init(N, A, B)` — El grader llamará esta función primeramente y exactamente una sola vez.

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

Usted 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` una vez, el grader llamara esta función  $Q$  veces en una fila, una vez por cada día.
- $M$ : el número de proyectos para este día.
- $K$ : un arreglo de longitud  $M$  conteniendo el tamaño del equipo requerido para cada uno de estos proyectos.
- La función debe retornar 1 si es posible formar todos los equipos requeridos y 0 en caso contrario.
- Usted puede asumir que  $1 \leq M \leq N$ , y que para cada  $i = 0, \dots, M - 1$  tenemos  $1 \leq K[i] \leq N$ . Note que la suma de todos los  $K[i]$  puede exceder  $N$ .

## Subtareas

Denote por  $S$  la suma de los valores de  $M$  en todas las llamadas a  $\text{can}(M, K)$ .

subtarea	puntos	$N$	$Q$	Restricciones adicionales
1	21	$1 \leq N \leq 100$	$1 \leq Q \leq 100$	none
2	13	$1 \leq N \leq 100,000$	$Q = 1$	none
3	43	$1 \leq N \leq 100,000$	$1 \leq Q \leq 100,000$	$S \leq 100,000$
4	23	$1 \leq N \leq 500,000$	$1 \leq Q \leq 200,000$	$S \leq 200,000$

## Sample grader

El ejemplo de grader lee la entrada en el siguiente formato:

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

Para cada pregunta, el ejemplo de grader imprime el valor retornado de `can`.