

## Montaña Rusa

Anna trabaja en un parque de diversiones, y tiene a cargo la construcción de las vías de una nueva montaña rusa. Ya ha diseñado  $n$  tramos especiales (convenientemente numerados de  $0$  a  $n - 1$ ) que afectan a la velocidad del carrito. Ahora debe ensamblarlos a todos para formar un recorrido, y así proponer el diseño final de la montaña rusa. A los efectos de este problema, se puede asumir que el carrito tiene longitud cero.

Para cada  $i$  entre  $0$  y  $n - 1$ , inclusive, el tramo especial  $i$  tiene dos propiedades:

- al entrar al mismo, hay un límite de velocidad: la velocidad del tren debe ser **menor o igual que**  $s_i$  km/h (kilómetros por hora),
- al salir del mismo, la velocidad del tren es de **exactamente**  $t_i$  km/h, sin importar la velocidad del tren al inicio del tramo.

La montaña rusa terminada consiste de una única vía recta, que contiene los  $n$  tramos especiales en algún orden. Cada uno de los  $n$  tramos debe ser usado exactamente una vez. Dos tramos especiales consecutivos se conectan entre sí usando un tramo normal de vías, de cierta longitud. Anna debe elegir el orden de los  $n$  tramos especiales, y luego decidir las longitudes de los tramos normales entre ellos. La longitud de un tramo normal se mide en metros, y puede ser igual a cualquier entero no-negativo (posiblemente cero).

Cada metro del tramo normal entre dos tramos especiales frena al tren en  $1$  km/h. Al comenzar el viaje en la montaña rusa, el carrito entra al primer tramo especial en el orden elegido por Anna, a una velocidad inicial de  $1$  km/h.

El diseño final debe satisfacer los siguientes requisitos:

- el carrito no viola ningún límite de velocidad al entrar a un tramo especial;
- la velocidad del carrito es positiva en todo momento.

En todas las subtareas excepto la subtarea 3, debe encontrar la mínima longitud total posible de los tramos normales. En la subtarea 3, solamente se debe verificar si es posible construir una montaña rusa válida tal que cada tramo normal tenga longitud cero.

### Detalles de implementación

Debe implementar la siguiente función:

- `int64 plan_roller_coaster(int[] s, int[] t)`

- **s**: arreglo de tamaño  $n$ , máximas velocidades de ingreso a cada tramo especial que están permitidas.
- **t**: arreglo de tamaño  $n$ , las velocidades de salida de cada tramo especial.
- En todas las subtareas excepto la subtarea 3, la función debe retornar la mínima longitud total posible de los tramos normales. En la subtarea 3, la función debe retornar **0** si existe una montaña rusa válida en la cual todos los tramos normales tengan longitud cero, o bien cualquier entero positivo si no existe tal montaña rusa.

Para el lenguaje C, la signatura de la función es levemente diferente:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
  - **n**: la cantidad de elementos en **s** y en **t** (o sea, la cantidad de tramos especiales),
  - los demás parámetros son iguales que antes.

## Ejemplo

`plan_roller_coaster([1, 4, 5, 6], [7, 3, 8, 6])`

En este ejemplo hay cuatro tramos especiales. La mejor solución es construirlos en el orden **0,3,1,2**, y conectarlos con tramos normales de longitudes **1,2,0** respectivamente. Así es como un carrito viaja por esta montaña rusa:

- La velocidad inicial es **1** km/h.
- El carrito entra por el tramo especial **0**.
- El carrito sale del tramo especial **0** a una velocidad de **7** km/h.
- Luego recorre un tramo normal de **1** m. Al terminar este tramo, su velocidad es de **6** km/h.
- El carrito entra al tramo especial **3**, a una velocidad de **6** km/h y sale del mismo con igual velocidad.
- Luego de salir del tramo especial **3**, el carrito viaja por un tramo normal de **2** m de longitud. Su velocidad se reduce hasta **4** km/h.
- El carrito ingresa al tramo especial **1**, con una velocidad de **4** km/h y sale a una velocidad de **3** km/h.
- Inmediatamente después de salir del tramo especial **1**, el carrito ingresa al tramo especial **2**.
- El tren sale del tramo especial **2**. Su velocidad final es **8** km/h.

La función debe retornar la longitud total de tramos normales en el recorrido:

$$1 + 2 + 0 = 3.$$

## Subtareas

En todas las subtareas,  $1 \leq s_i \leq 10^9$  y  $1 \leq t_i \leq 10^9$ .

1. (11 puntos):  $2 \leq n \leq 8$ ,

2. (23 puntos):  $2 \leq n \leq 16$ ,
3. (30 puntos):  $2 \leq n \leq 200\,000$ . En esta subtarea su programa solo debe verificar si la respuesta es cero o no. Si la respuesta no es cero, cualquier entero positivo se considera correcto.
4. (36 puntos):  $2 \leq n \leq 200\,000$ .

### Evaluador de ejemplo

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

- línea 1: un entero  $n$ .
- línea  $2 + i$ , para cada  $i$  entre  $0$  y  $n - 1$ : enteros  $s_i$  y  $t_i$ .