

## Roller Coaster Railroad

Anna está trabajando en un parque de diversiones y está a cargo de construir los diseños de los carriles para una nueva montaña rusa. Ella ya ha diseñado  $n$  secciones especiales. ( convenientemente enumeradas de  $0$  a  $n - 1$  ) que afectan a la velocidad de un tren de la montaña rusa como: colinas, pistas de freno, y otros. Ahora ella tiene que ponerlos juntos y proponer un diseño final para la montaña rusa.

Para cada  $i$  comprendido entre  $0$  y  $n - 1$  , inclusive, la sección especial  $i$  tiene dos propiedades:

- cuando se ingresa a la sección, existe un límite de velocidad: la velocidad del tren debe ser **al menos (menor o igual a)**  $s_i$  km/h (kilometros por hora),
- cuando se abandona la sección, la velocidad del tren es **exactamente**  $t_i$  km/h, independientemente a la velocidad con la cual ingreso el tren a la sección.

La montaña rusa concluida debe contener cada uno de las  $n$  secciones exactamente una vez. Sin embargo, debe haber un riel entre cada dos secciones consecutivas. Anna debe escoger el orden de las  $n$  secciones y decidir las longitudes de cada riel. La longitud de un riel se mide en metros y debe ser igual a cualquier número entero no negativo (Posiblemente zero)

Cada metro de la via entre dos secciones especiales reduce la velocidad del tren por 1 km/h. Al principio del recorrido, el tren ingresa a la primera seccion especial a una velocidad de 1 km/h.

El diseño final debe satisfacer los siguientes requerimientos:

- el tren no debe violar ningun límite de velocidad mientras está ingresando a la sección especial;
- la velocidad del tren es positiva en cualquier momento hasta que alcanza la parte final de la última sección especial.

En todas las subtareas excepto la subtarea 3, su labor es encontrar el mínimo largo total posible de las vías entre las secciones. En la subtarea 3 solo necesita revisar si existe un diseño de montaña rusa válido, de modo que cada vía tenga largo cero.

### Detalles de implementación

Debes implementar las siguientes funciones (method):

- `int64 plan_roller_coaster(int[] s, int[] t)`
  - $s$ : vector de longitud  $n$  , entrada de velocidades máxima permitida.
  - $t$ : vector de longitud  $n$  , velocidades de salida.

- La función debe retornar la longitud posible total mínima de todos los caminos entre las secciones especiales (en la subtarea 3, puedes sacar de salidad un número entero positivo si la respuesta no es cero, observa el detalle en la sección de la subtarea).

Para el lenguaje C, el esquema de la función es un tanto diferente:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
  - `n`: el número de elementos en `s` y `t` (es decir, el número de secciones especiales),
  - los otros parametros son los mismo que de los de arriba.

## Ejemplo

`int64 plan_roller_coaster([1, 4, 5, 6], [7, 3, 8, 6])`

En este ejemplo existe cuatro secciones especiales. La mejor solución es construirlas en el siguiente orden `0,3,1,2`, y conectarlas por caminos de longitudes `1,2,0` respectivamente. De esta forma el tren recorre por todo el camino de ferrocarril :

- Inicialmente la velocidad del tren es `1` km/h.
- El tren empieza el recorrido entrando a la sección `0`.
- El tren deja la sección `0` a una velocidad de `7` km/h.
- Posteriormente hay una pista de longitud de `1` m. Cuando el tren alcanza el final del camino, su velocidad es `6` km/h.
- El tren ingresa a la sección especial `3` a `6` km/h y la abandona a la misma velocidad.
- Después de abandonar la sección `3`, el tren viaja a través de una sección de `2` m. Su velocidad se reduce a `4` km/h.
- El tren ingresa a la sección especial `1` a `4` km/h y la abandona a `3` km/h.
- Inmediatamente después de la sección `1` el tren ingresa a la sección especial `2`.
- El tren abandona la sección `2`. Su velocidad final es `8` km/h.

La función debe retornar la longitud total del camino entre las secciones especiales  $1 + 2 + 0 = 3$ .

## 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 estas subtarea tu programa solo necesita verificar si la respuesta es cero o no. Si la respuesta no es cero, cualquier número entero positivo es considerado correcto.
4. (36 puntos):  $2 \leq n \leq 200\,000$ .

## Sample grader

The sample grader reads the input in the following format:

- línea 1: entero  $n$ .
- línea  $2 + i$ , por cada  $i$  comprendida entre  $0$  and  $n - 1$ : enteros  $s_i$  y  $t_i$ .