



Roller Coaster Railroad

L'Anna treballa en un parc d'atraccions i s'encarrega de construir les vies d'una nova muntanya russa. Ja ha dissenyat n seccions especials (numerades de la 0 a la $n - 1$) que afecten la velocitat d'un tren. Ara ha de posar-les juntes i proposar un disseny final de muntanya russa. En aquest problema podeu suposar que la mida del tren és zero.

Per a cada i entre 0 i $n - 1$, inclosos, la secció especial i té dues propietats:

- quan s'entra a la secció, hi ha un límit de velocitat: la velocitat del tren ha de ser **menor o igual** que s_i km/h (kilometres per hora),
- quan se surt d'una secció, la velocitat del tren és **exactament** t_i km/h, sense importar la velocitat amb què el tren ha entrat a la secció.

La muntanya russa acabada és una única via que conté les n seccions en algun ordre. Cadascuna de les n seccions s'ha d'utilitzar exactament una vegada. Les seccions consecutives estan connectades mitjançant trams de via. L'Anna ha de triar l'ordre de les n seccions, i també ha de decidir la mida dels trams de via intermitjos. La llargada de cada tram es mesura en metres, i pot ser qualsevol enter no negatiu (possiblement zero).

Cada metre de via entre dues seccions especials alenteix el tren en 1 km/h. Al començament de l'atracció, el tren entra a la primera secció especial segons l'ordre escollit per l'Anna amb una velocitat d' 1 km/h.

El disseny final ha de satisfer els requeriments següents:

- el tren no pot superar el límit de velocitat quan entra a les seccions especials;
- la velocitat del tren ha de ser positiva en tot moment.

En totes les subtasques, llevat de la subtasca 3, la teva tasca consisteix a trobar la mínima llargada total possible dels trams de via entre les seccions. En la subtasca 3 només heu de comprovar si existeix un disseny de muntanya russa vàlid tal que tots els trams de via intermitjos tinguin llargada zero.

Detalls d'implementació

Se us demana que implementeu una funció (mètode):

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : array de mida n , màximes velocitats d'entrada permeses.
 - t : array de mida n , velocitats de sortida.

- En totes les subtasques, llevat de la subtasca 3, la funció ha de retornar la mínima llargada total possible de tots els trams de via entre seccions especials. En la subtasca 3 la funció ha de retornar **0** si existeix un disseny de muntanya russa vàlid tal que cada tram de via tingui llargada zero, i qualsevol enter positiu si no existeix.

La capçalera de la funció és lleugerament diferent en el llenguatge C:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - **n**: el nombre d'elements a **s** i **t** (és a dir, el nombre de seccions especials),
 - la resta de paràmetres són els mateixos que abans.

Exemple

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

En aquest exemple hi ha quatre seccions especials. La solució òptima consisteix a construir-los en l'ordre **0,3,1,2**, i connectar-los amb trams de vies de llargada **1,2,0** respectivament. El recorregut que faria el tren seria el següent:

- La velocitat inicial del tren és d' **1** km/h.
- El tren comença l'atracció quan entra a la secció especial **0**.
- El tren surt de la secció **0** a una velocitat de **7** km/h.
- A continuació, hi ha un tram de via de llargada **1** m. Quan el tren arriba a la fi del tram, la seva velocitat és de **6** km/h.
- El tren entra a la secció especial **3** amb una velocitat de **6** km/h, i en surt a la mateixa velocitat.
- Després de sortir de la secció **3**, el tren recorre un tram de via de **2** m. La seva velocitat es redueix fins als **4** km/h.
- El tren entra a la secció especial **1** amb una velocitat de **4** km/h, i en surt a una velocitat de **3** km/h.
- El tren entra a la secció especial **2** immediatament després de sortir de la secció especial **1**.
- El tren surt de la secció **2**. La seva velocitat final és de **8** km/h.

La funció ha de retornar la llargada total dels trams de via entre seccions especials: $1 + 2 + 0 = 3$.

Subtasques

Per a totes les subtasques es compleix $1 \leq s_i \leq 10^9$ i $1 \leq t_i \leq 10^9$.

1. (11 punts): $2 \leq n \leq 8$,
2. (23 punts): $2 \leq n \leq 16$,
3. (30 punts): $2 \leq n \leq 200\,000$. En aquesta subtasca el teu programa només ha de comprovar si la resposta és zero o no. Si la resposta no és zero, qualsevol enter positiu es considera correcte.

4. (36 punts): $2 \leq n \leq 200\,000$.

Grader de mostra

El grader de mostra llegeix l'entrada en el format següent:

- línia 1: enter n .
- línia $2 + i$, per a i entre 0 i $n - 1$: enters s_i i t_i .