

Trilhos da Montanha-Russa

Anna está trabalhando num parque de diversões e foi encarregue de construir uma nova montanha-russa. Ela já elaborou n seções especiais (convenientemente numeradas de 0 a $n - 1$) que afetam a velocidade de um carrinho na montanha-russa. Agora ela precisa uní-las para propor a configuração final da montanha-russa. Para os propósitos dessa tarefa você pode supor que o comprimento de um carrinho é zero.

Para cada i entre 0 e $n - 1$, inclusive, a seção especial i tem duas propriedades:

- ao entrar na seção, há um limite de velocidade: a velocidade do carrinho deve ser **menor do que ou igual a** s_i km/h (quilômetros por hora),
- ao sair da seção, a velocidade do carrinho será **exatamente** t_i km/h, independentemente da velocidade com a qual o carrinho entrou na seção.

A montanha-russa finalizada é uma linha férrea que contém as n seções especiais em alguma ordem. Cada uma das n seções deve ser usada exatamente uma vez. Seções especiais consecutivas são conectadas por trilhos. Anna deve escolher a ordem das n seções e depois decidir o comprimento de cada trilho. O comprimento de um trilho é medido em metros e pode ser igual a qualquer inteiro não negativo (possivelmente zero).

Cada metro de trilho entre duas seções especiais desacelera o carrinho em 1 km/h. No início do trajeto, o carrinho entra na primeira seção especial da ordem escolhida por Anna com velocidade de 1 km/h.

A configuração final deve satisfazer os seguintes requisitos:

- o carrinho não viola os limites de velocidade ao entrar nas seções especiais;
- a velocidade do carrinho é positiva a todo momento.

Em todas as subtarefas com exceção da subtarefa 3, sua tarefa é encontrar o menor comprimento total possível dos trilhos entre as seções. Na subtarefa 3 você só precisa verificar se existe uma configuração válida para a montanha-russa em que cada trilho tem comprimento nulo.

Detalhes da implementação

Você deve implementar a seguinte função (método):

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : vetor de tamanho n , velocidades máximas de entrada permitidas.
 - t : vetor de tamanho n , velocidades de saída.

- Em todas as subtarefas com exceção da subtarefa 3, a função deve retornar o menor comprimento total possível de todos os trilhos. Na subtarefa 3, a função deve retornar **0** se existir uma configuração final válida na qual todos os trilhos têm comprimento zero, e qualquer inteiro positivo se tal configuração não existir.

Para a linguagem C o protótipo da função é um pouco diferente:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - **n**: o número de elementos em **s** e **t** (ou seja, o número de seções especiais),
 - os outros parâmetros são iguais aos acima.

Exemplo

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

Neste exemplo há quatro seções especiais. A melhor solução é construí-las na ordem **0,3,1,2**, e conectá-las com trilhos de comprimentos **1,2,0** respectivamente. Um carrinho atravessa esta montanha russa da seguinte forma:

- Inicialmente a velocidade do carrinho é **1** km/h.
- O carrinho começa o percurso entrando na seção especial **0**.
- O carrinho sai da seção **0** a **7** km/h.
- Em seguida existe um trilho de **1** m. Quando o carrinho chega no final do trilho sua velocidade é **6** km/h.
- O carrinho entra na seção especial **3** a **6** km/h e sai com a mesma velocidade.
- Após deixar a seção **3**, o carrinho passa por um trilho de **2** m de comprimento. Sua velocidade diminui para **4** km/h.
- O carrinho entra na seção especial **1** a **4** km/h e sai a **3** km/h.
- Imediatamente após a seção especial **1** o carrinho entra na seção especial **2**.
- O carrinho deixa a seção especial **2**. Sua velocidade final é **8** km/h.

A função deve retornar o comprimento total dos trilhos entre seções especiais:

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

Subtarefas

Em todas as subtarefas $1 \leq s_i \leq 10^9$ e $1 \leq t_i \leq 10^9$.

1. (11 pontos): $2 \leq n \leq 8$,
2. (23 pontos): $2 \leq n \leq 16$,
3. (30 pontos): $2 \leq n \leq 200\,000$. Nesta subtarefa seu programa só precisa verificar se a resposta é zero ou não. Se a resposta não é zero, qualquer inteiro positivo é considerado uma resposta correta.
4. (36 pontos): $2 \leq n \leq 200\,000$.

Corretor exemplo

O corretor exemplo lê a entrada no seguinte formato:

- linha 1: inteiro n .
- linha $2 + i$, para i entre 0 e $n - 1$: inteiros s_i e t_i .