



Wiring

Maryam é uma engenheira elétrica. Ela está a projetar o cablamento numa torre de comunicação. Na torre existem alguns pontos de conexão posicionados em alturas distintas. Um cabo pode ser usado para conectar quaisquer dois pontos de conexão. Cada ponto de conexão pode ser conectado a um número arbitrário de cabos. Existem dois tipos de pontos de conexão: vermelho e azul.

Para o propósito deste problema a torre deve ser vista como uma linha e os pontos de conexão como pontos azuis e vermelhos que estão em coordenadas inteiras não-negativas nesta linha. O comprimento de um cabo é a distância entre os dois pontos de conexão que ele conecta.

O seu objetivo é ajudar a Maryam a encontrar um esquema de cablamento tal que:

1. Cada ponto de conexão tem pelo menos um cabo para um ponto de conexão de uma cor diferente.
2. O comprimento total dos cabos é minimizado.

Detalhes de implementação

Deve implementar a seguinte função:

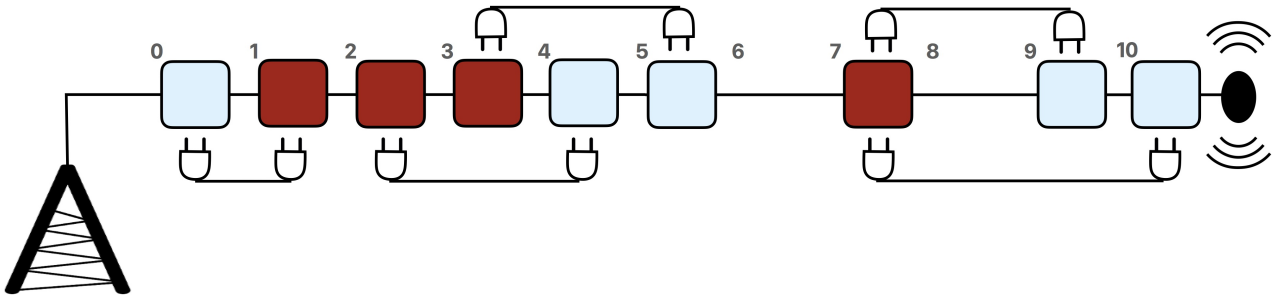
```
int64 min_total_length(int[] r, int[] b)
```

- r : vetor de comprimento n contendo as posições dos pontos de conexão vermelhos por ordem crescente.
- b : vetor de comprimento m contendo as posições dos pontos de conexão azuis por ordem crescente.
- Esta função deve retornar o comprimento total mínimo dos cabos, entre todos os esquemas de cablamento válidos.
- Note que o tipo de retorno desta função é `int64`.

Exemplo

```
min_total_length([1, 2, 3, 7], [0, 4, 5, 9, 10])
```

A figura seguinte ilustra este exemplo.



- A torre é mostrada horizontalmente.
- Na versão de impressão a preto-e-branco do enunciado do problema os pontos de conexão vermelhos são escuros e os azuis são claros.
- Existem 4 pontos de conexão vermelhos, localizados nas posições 1, 2, 3 e 7.
- Existem 5 pontos de conexão azuis, localizados nas posições 0, 4, 5, 9 e 10.
- Uma solução ótima é mostrada na figura acima.
- Nesta solução, o comprimento total dos cabos é $1 + 2 + 2 + 2 + 3 = 10$, que é ótima. Então, a função deve retornar 10.
- Note que dois cabos estão conectados ao ponto de conexão na posição 7.

Restrições

- $1 \leq n, m \leq 100\,000$,
- $0 \leq r[i] \leq 10^9$ (para todo $0 \leq i \leq n - 1$),
- $0 \leq b[i] \leq 10^9$ (para todo $0 \leq i \leq m - 1$),
- Cada um dos vetores r e b está ordenado por ordem crescente.
- Todos os $n + m$ valores nos vetores r e b são distintos.

Subtarefas

1. (7 pontos) $n, m \leq 200$,
2. (13 pontos) Todos os pontos vermelhos têm posições menores do que qualquer ponto azul.
3. (10 pontos) Existe pelo menos um ponto de conexão vermelho e um ponto de conexão azul entre cada 7 pontos de conexão sucessivos.
4. (25 pontos) Todos os pontos de conexão têm posições diferentes no intervalo $[1, n + m]$.
5. (45 pontos) Nenhuma restrição adicional.

Avaliador exemplo

O avaliador exemplo lê o *input* no seguinte formato:

- linha 1: $n \ m$
- linha 2: $r[0] \ r[1] \ \dots \ r[n - 1]$
- linha 3: $b[0] \ b[1] \ \dots \ b[m - 1]$

O avaliador exemplo imprime uma única linha contendo o valor de retorno de `min_total_length`.