



Cablaggio

Maryam, ingegnere elettrico, è stata incaricata di cablare una torre di comunicazione. Lungo la torre ci sono un certo numero di connettori, disposti ad altezze diverse. I cavi possono essere usati per collegare una qualsiasi coppia di connettori, e a ciascun connettore si possono collegare un numero arbitrario di cavi. Ci sono due tipi di connettori: rossi e blu.

Ai fini di questo problema, la torre può essere vista come una linea retta, ed i connettori come dei punti blu e rossi che si trovano a coordinate non-negative di questa linea. La lunghezza di un cavo è data dalla distanza tra i due connettori che esso collega.

Il tuo obiettivo è quello di aiutare Maryam progettando un cablaggio tale che:

1. ogni connettore è collegato ad almeno un connettore di colore diverso;
2. la lunghezza totale dei cavi usati è minimizzata.

Dettagli di implementazione

Devi implementare la seguente funzione:

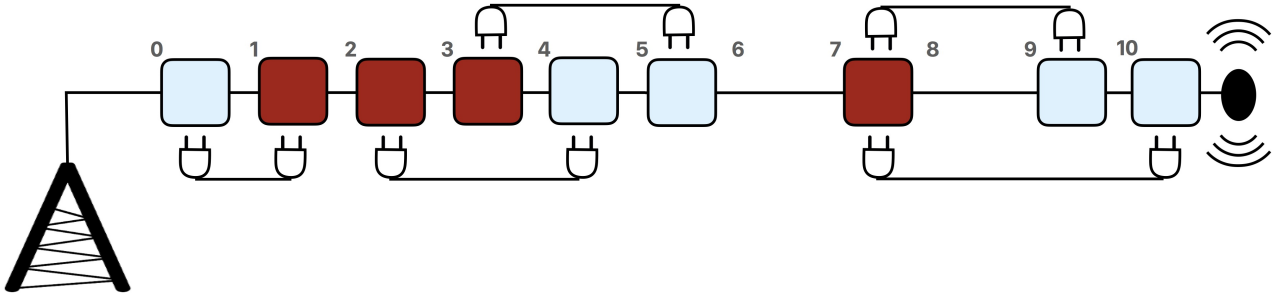
```
int64 min_total_length(int[] r, int[] b)
```

- r : un array di lunghezza n che contiene le posizioni dei connettori rossi, in ordine crescente.
- b : un array di lunghezza m che contiene le posizioni dei connettori blu, in ordine crescente.
- La funzione deve restituire la minima lunghezza totale dei cavi, tra tutte le possibili configurazioni di cablaggio.
- Nota che il tipo di ritorno di questa funzione è `int64`.

Esempio

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

La figura qui sotto illustra l'esempio.



- La torre è mostrata in orizzontale.
- Nella versione del testo stampata in bianco e nero i connettori rossi sono quelli scuri mentre i connettori blu sono quelli chiari.
- Ci sono 4 connettori rossi nelle posizioni 1, 2, 3, e 7.
- Ci sono 5 connettori blu nelle posizioni 0, 4, 5, 9, e 10.
- La figura sopra mostra una possibile soluzione ottima.
- In questa soluzione, la lunghezza totale dei cavi è $1 + 2 + 2 + 2 + 3 = 10$, che è ottimo quindi la funzione deve restituire 10.
- Nota che al connettore in posizione 7 sono stati collegati due cavi.

Assunzioni

- $1 \leq n, m \leq 100\,000$,
- $0 \leq r[i] \leq 10^9$ ($0 \leq i \leq n - 1$),
- $0 \leq b[i] \leq 10^9$ ($0 \leq i \leq m - 1$),
- Gli array r e b sono ordinati in modo crescente.
- Tutti gli $n + m$ valori negli array r e b sono distinti.

Assegnazione del punteggio

1. **(7 punti)** $n, m \leq 200$,
2. **(13 punti)** Tutti i connettori rossi si trovano in posizioni minori (più a sinistra) dei connettori blu.
3. **(10 punti)** Per ogni 7 connettori consecutivi, ci sono sempre almeno un connettore rosso ed almeno un connettore blu.
4. **(25 punti)** Tutti i connettori hanno posizioni distinte comprese nel range $[1, n + m]$.
5. **(45 punti)** Nessuna limitazione specifica.

Grader di prova

Il grader di prova legge l'input nel seguente formato:

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

Il grader di prova stampa una sola riga contenente il valore restituito da `min_total_length`.