



雲霄飛車鐵道 Roller Coaster Railroad

安娜在遊樂園工作並負責建造一個新的雲霄飛車鐵道。她已經設計好 n 個特別區段(編號0至 $n - 1$)，來改變雲霄飛車的速度。她現在必須把這些區段組合起來，以提出雲霄飛車的最終設計。在此題目中，你可以假設車子的長度是零。

對介於0至 $n - 1$ 的 i ，特別區段 i 有兩個特性：

- 進入區段時有速限：飛車的速度必須小於或等於 s_i km/h (時速 s_i 公里)。
- 離開區段時，飛車的速度剛剛好會變成是 t_i km/h，不管飛車進入區段時速度為何。

完成後的雲霄飛車是個單一線型的鐵道線，包含某個順序的 n 個特別區段，每個區段都剛好只用到一次，而且兩個區段之間必須有連結軌道。安娜必須選擇 n 個區段的順序，然後決定每段連結軌道的長度。連結軌道的長度以公尺表示，可以是任何非負數的整數(可能是零)。

兩個特別區段之間，每一公尺的連結軌道會導致飛車的速度降低 1 km/h (時速1公里)。在雲霄飛車的起點，飛車進入安娜所選的第一個特別區段的速度是 1 km/h。

最終設計必須滿足以下條件：

- 飛車進入特別區段時，不可違反速限；
- 飛車的速度在任何時候都為正數。

在全部子任務中，第三個子任務除外，你的任務是找出 n 個特別區段的順序，以及區段之間連結軌道的長度，使得連結軌道的長度總和愈小愈好。在第三個子任務中，你只要檢驗是否存在每個連結軌道長度都是零的有效雲霄飛車設計。

實作細節

你必須實作此函式(或方法)：

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : 長度 n 的陣列，進入時可允許的最高速度。
 - t : 長度 n 的陣列，離開時的速度。
 - 在全部子任務中，第三個子任務除外，函式必須回傳最小可能的區段間連結軌道長度總和。在第三個子任務中，如果存在每個連結軌道長度都是零的有效雲霄飛車設計，則函式必須回傳零。如果這樣的設計不存在，則函式可以回傳任何正整數。

若使用C程式語言，函式參數稍有不同：

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - n : the number of elements in s 和 t 陣列元素個數(也就是特別區段的個數)，
 - 其它參數同上。

範例

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

此範例有四個是特別區段。最佳解是以0, 3, 1, 2的順序建造，再個別使用長度1, 2, 0的連結軌道。

於是飛車在鐵道上運行的方式會是：

- 飛車起始速度是 1 km/h。
- 飛車首先進入特別區段 0。
- 飛車離開區段 0 時的速度為 7 km/h。
- 然後是長度 1 公尺的連結軌道，當飛車走完連結軌道時速度成為 6 km/h。
- 飛車進入特別區段 3 時的速度為 6 km/h，且離開時速度不變。
- 離開區段 3 後，飛車走上一段 2 公尺的連結軌道，它的速度降低至 4 km/h。
- 飛車進入特別區段 1 時的速度為 4 km/h，且離開時速為 3 km/h。
- 離開特別區段 1 後，飛車立即進入特別區段 2。
- 飛車離開區段 2，最終速度為 8 km/h。

函式必須回傳區段間連結軌道的長度總和： $1 + 2 + 0 = 3$ 。

子任務(Subtasks)

在全部子任務中， $1 \leq s_i \leq 10^9$ 且 $1 \leq t_i \leq 10^9$ 。

1. (11 points): $2 \leq n \leq 8$,
2. (23 points): $2 \leq n \leq 16$,
3. (30 points): $2 \leq n \leq 200\,000$. 在這個子任務中，你的程式只要檢查答案是否為零。如果答案不是零，任何正整數都視為正確。
4. (36 points): $2 \leq n \leq 200\,000$.

範例評分程式(Sample grader)

範例評分程式用下列格式讀取輸入資料：

- 第 1 行：整數 n 。
- 第 $2 + i$ 行， i 介於 0 至 $n - 1$ 之間：整數 s_i 和 t_i 。