



Đường ray cho tàu lượn cao tốc (Roller Coaster Railroad)

Anna đang làm việc cho công viên vui chơi giải trí và nhận nhiệm vụ xây dựng đường ray cho tàu lượn cao tốc mới. Cô ấy đã thiết kế n đoạn đường đặc biệt (để thuận tiện sẽ được đánh số từ 0 đến $n - 1$) có tác động đến tốc độ của tàu lượn cao tốc. Bây giờ cô ấy cần ghép chúng lại với nhau để đưa ra bản thiết kế đường ray cuối cùng cho tàu lượn cao tốc. Với mục đích của bài này bạn có thể coi độ dài của tàu là bằng không.

Với mỗi i trong khoảng 0 và $n - 1$, kể cả hai đầu mút, đoạn đường i có hai tính chất:

- khi đi vào đoạn đường, có hạn chế về tốc độ: tốc độ của tàu phải **nhỏ hơn hoặc bằng** s_i km/h (kilomet/giờ),
- khi đi ra khỏi đoạn đường, tốc độ của tàu **đúng bằng** t_i km/h, không phụ thuộc vào tốc độ của tàu khi đi vào đoạn đường.

Bản thiết kế cuối cùng cho tàu lượn cao tốc là một đường ray gồm n đoạn đường đặc biệt theo một trình tự nào đó. Mỗi một trong số n đoạn đường phải được sử dụng đúng một lần. Các đoạn đường liên tiếp được nối với nhau bởi đoạn ray. Anna phải chọn thứ tự n đoạn đường và sau đó quyết định độ dài của mỗi đoạn ray nối. Độ dài của đoạn ray nối được tính bởi mét và có thể là một số nguyên không âm bất kỳ (có thể bằng không).

Mỗi mét đoạn ray nối giữa hai đoạn đường đặc biệt làm tốc độ của tàu giảm đi 1 km/h. Tại lúc bắt đầu chạy, tàu lượn cao tốc đi vào đoạn đường đặc biệt đầu tiên trong thứ tự được chọn bởi Anna với tốc độ 1 km/h.

Bản thiết kế cuối cùng phải thoả mãn các đòi hỏi sau đây:

- tàu lượn không được vi phạm hạn chế về tốc độ khi đi vào các đoạn đường đặc biệt;
- tốc độ của tàu là số dương tại mọi thời điểm.

Trong tất cả các subtasks trừ subtask 3, bạn có nhiệm vụ tìm giá trị nhỏ nhất có thể của tổng các độ dài của các đoạn ray nối giữa các đoạn đường liên tiếp. Trong subtask 3 bạn chỉ cần kiểm tra xem liệu có tồn tại cách thiết kế đường ray tàu lượn cao tốc sao cho mỗi đoạn ray nối đều có độ dài là 0 hay không.

Chi tiết cài đặt

Bạn phải cài đặt hàm (method) sau:

- `int64 plan_roller_coaster(int[] s, int[] t)`

- **s**: mảng độ dài n , giới hạn tối đa cho phép cho tốc độ đi vào.
- **t**: mảng độ dài n , tốc độ đi ra.
- Trong tất cả các subtasks trừ subtask 3, hàm phải trả lại tổng độ dài nhỏ nhất có thể của tất cả các đoạn ray nối. Trong subtask 3 hàm phải trả lại **0** nếu tồn tại cách thiết kế đường ray tàu lượn cao tốc mà mỗi đoạn ray nối đều có độ dài là **0** và đưa ra số nguyên dương tùy ý nếu như cách thiết kế như vậy là không tồn tại.

Đối với ngôn ngữ C khuôn dạng của hàm hơi khác một chút:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - **n**: số lượng phần tử trong **s** và **t** (là số lượng đoạn đường đặc biệt),
 - các tham số khác có giống như đã mô tả ở trên.

Ví dụ

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

Trong ví dụ này có bốn đoạn đường đặc biệt. Lời giải tốt nhất là xây dựng đường ray theo thứ tự **0,3,1,2**, và nối chúng bởi các đoạn ray với độ dài tương ứng là **1,2,0**.

Dưới đây là cách tàu lượn đi qua hệ thống đường ray này:

- Đầu tiên tốc độ của tàu là **1** km/h.
- Tàu bắt đầu chạy bởi việc đi vào đoạn đường **0**.
- Tàu đi ra khỏi đoạn đường **0** với tốc độ **7** km/h.
- Tiếp đến có đoạn ray nối độ dài **1** m. Khi đi đến cuối đoạn ray này tốc độ của tàu là **6** km/h.
- Tàu đi vào đoạn đường **3** với tốc độ **6** km/h và đi ra khỏi nó với cùng tốc độ.
- Sau khi đi ra khỏi đoạn đường **3**, tàu đi theo **2** m đoạn ray nối. Tốc độ của tàu giảm xuống còn **4** km/h.
- Tàu đi vào đoạn đường **1** với tốc độ **4** km/h và đi ra khỏi nó với tốc độ **3** km/h.
- Ngay sau khi đi ra khỏi đoạn đường **1** tàu đi vào đoạn đường **2**.
- Tàu đi ra khỏi đoạn đường **2**. Tốc độ cuối cùng của tàu là **8** km/h.

Hàm cần trả lại tổng độ dài của các đoạn ray nối giữa các đoạn đường là:

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

Subtasks

Trong tất cả các subtasks $1 \leq s_i \leq 10^9$ và $1 \leq t_i \leq 10^9$.

- (11 points): $2 \leq n \leq 8$,
- (23 points): $2 \leq n \leq 16$,
- (30 points): $2 \leq n \leq 200\,000$. Trong subtask này chương trình của bạn chỉ cần kiểm tra xem câu trả lời có phải bằng **0** hay không. Nếu câu trả lời không phải là **0**, thì mọi số nguyên dương đều được coi là câu trả lời đúng.
- (36 points): $2 \leq n \leq 200\,000$.

Chương trình chấm mẫu (Sample grader)

Chương trình chấm mẫu đọc dữ liệu vào theo khuôn dạng:

- dòng 1: số nguyên n .
- dòng $2 + i$, với mỗi i trong khoảng giữa 0 và $n - 1$: các số nguyên s_i và t_i .