



## 过山车铁路

Anna 在一个游乐园工作。她负责建造一个新的过山车铁路。她已经设计了影响过山车速度的  $n$  个特殊的路段(方便起见标记为  $0$  到  $n-1$ )。现在 Anna 必须要把这些特殊的路段放在一起并提出一个过山车的最后设计。为了简化问题,你可以假设过山车的长度为零。

对于  $0$  和  $n-1$  之间的每个  $i$ , 这个特殊的路段  $i$  具有如下两个性质:

- 当进入这个路段时, 有一个速度限制: 过山车的速度必须小于或等于  $s_i$  km/h (每小时千米),
- 当离开这个路段时, 过山车的速度刚好是  $t_i$  km/h, 不管过山车进入该路段时的速度如何。

最后完成的过山车设计是一个以某种顺序包含这  $n$  个特殊路段的单一铁路线。这  $n$  个路段中的每一个应当被使用刚好一次。连续的路段之间用铁轨来连接。Anna 应该选择这  $n$  个路段的顺序, 然后确定每段铁轨的长度。铁轨的长度以米来度量, 可以是任意的非负整数(可以为零)。

两个特殊路段之间的每 1 米铁轨可以将过山车的速度减慢 1 km/h。在这个过山车铁路的起点, 过山车按照 Anna 选择的顺序进入第一个特殊路段时的速度是 1 km/h。

最后的设计还必须满足以下需求:

- 过山车在进入这些特殊路段时不能违反任何一个速度限制;
- 过山车的速度在任意时刻均为正。

在所有子任务中(子任务 3 除外), 你的任务是找出这些路段之间铁轨的最小可能总长度(这些路段之间铁轨总长度的最小值)。对于子任务 3 你只需要检查是否存在一个有效的过山车设计, 使得每段铁轨的长度均为零。

### 实现细节

你应该实现以下函数(方法):

- `int64 plan_roller_coaster(int[] s, int[] t)`
  - $s$ : 长度为  $n$  的数组, 进入路段时允许的速度最大值。
  - $t$ : 长度为  $n$  的数组, 离开路段时的速度。
  - 在所有子任务中(子任务 3 除外), 这个函数应该返回所有铁轨的最小可能的总长度。(在子任务 3 中, 如果存在一个有效的过山车设计使得每段铁轨的长度均为零, 则函数返回零, 如果上述设计不存在, 则输出任意的正整数)。

对于 C 语言, 这个函数的参数略有不同:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
  - $n$ :  $s$  和  $t$  中元素的个数(即, 特殊路段的数目),
  - 其他参数同上。

## 样例

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

在这个样例中有4个特殊的路段。最好的解是按照 0,3,1,2 的顺序建造,连接这些路段的铁轨长度分别是 1,2,0。下面给出过山车沿铁路铁轨的行驶方式:

- 最初过山车的速度是 1 km/h。
- 过山车由进入 0 号路段开始行进。
- 过山车以 7 km/h 的速度离开 0 号路段。
- 然后有一段长度为 1 m 的铁轨。过山车在到达这段铁轨的末端时速度为 6 km/h。
- 过山车以 6 km/h 的速度进入 3 号路段并以相同的速度离开该路段。
- 在离开 3 号路段后,过山车走过一段 2 m 长的铁轨。速度降至 4 km/h。
- 过山车以 4 km/h 的速度进入 1 号路段,并且以 3 km/h 的速度离开该路段。
- 离开 1 号路段后,过山车立即进入 2 号路段。
- 过山车离开 2 号路段。其最终速度是 8 km/h。

这个函数应该返回路段之间的铁轨总长度:  $1 + 2 + 0 = 3$ 。

## 子任务

在所有子任务中  $1 \leq s_i \leq 10^9$  并且  $1 \leq t_i \leq 10^9$

1. (11分):  $2 \leq n \leq 8$ ,
2. (23分):  $2 \leq n \leq 16$ ,
3. (30分):  $2 \leq n \leq 200\,000$ 。在这个子任务中,你的程序仅仅需要检查答案是否为零。如果答案不为零,则任意的正整数答案均被认为是正确的。
4. (36分):  $2 \leq n \leq 200\,000$ 。

## 样例评测程序

样例测评程序按照以下格式读入输入:

- 第 1 行 整数  $n$ 。
- 第  $2 + i$  行,对于 0 和  $n - 1$  之间的每个  $i$ : 整数  $s_i$  和  $t_i$ 。