



"Amerika dağları" attraksionunun dəmiryolu

Anna əyləncə parkında işləyir və o yeni "Amerika dağları" attraksionu üçün dəmiryolunun tikintisinə məsuldur. O, artıq n sayda xüsusi bölümü (0 -dan $n - 1$ -dək ardıcıl nömrələnmiş) elə layihələndirib ki, attraksion qatarının sürətinə təsir edir. Anna indi həmin bölümləri bir yerdə gözdən keçirib attraksionun layihəsini bitirmək niyyətindədir. Burada qatarın uzunluğunun sıfır olmasını qəbul edin.

0 və $n - 1$ aralığında olan hər bir i üçün i xüsusi bölümü iki özəlliyə malikdir:

- bölümə daxil olarkən sürət həddi (limiti) var: qatarın sürəti ən çoxu s_i km/saat olmalıdır,
- bölümdən çıxarkən qatarın sürəti daxil olduğu sürətdən asılı olmayaraq, **dəqiq** t_i km/saat olur.

Yekunda attraksion müəyyən ardıcılıqla yerləşmiş n xüsusi bölümündən ibarət bir dəmiryoludur. Hər bir bölüm yalnız bir dəfə istifadə olunmalıdır. İstənilən iki ardıcıl bölüm yollarla birləşdirilib. Anna n bölümü hansı ardıcılıqla yerləşdirmək məsələsini həll etməli və sonra hər bir birləşdirici yolun uzunluğunu seçməlidir. Yolun uzunluğu metrle ölçülür və istənilən mənfi olmayan ədəd (o cümlədən sıfır) ola bilər.

İki xüsusi bölüm arasındakı yolun hər bir metri qatarı 1 km/saat yavaşdır. Gəzintinin başlanğıcında qatar birinci xüsusi bölümə 1 km/saat sürətlə daxil olur.

Son layihə aşağıdakı tələbləri ödəməlidir:

- qatar xüsusi bölümə daxil olarkən hər hansı sürət həddini pozmamalıdır;
- qatarın sürəti istənilən anda müsbətdir.

Bütün altməsələlərdə (3-cüdən başqa) siz bölümlər ardıcılığını və ardıcıl bölümlər arasındakı yolların uzunluqlarını tapmalısınız, belə ki, bölümlər arasındakı yolların toplam uzunluğu mümkün qədər kiçik olsun. 3-cü altməsələdə isə siz yalnız hər bir yolun uzunluğu sıfır olan "Amerika dağları" attraksionunu layihələndirməyin mümkün olub-olmadığını yoxlamalısınız.

Implementation details

You should implement the following function (method):

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : array of length n , maximum allowed entry speeds.
 - t : array of length n , exit speeds.
 - The function should return the minimum possible total length of all tracks

between special sections (in subtask 3 you can output any positive integer if answer is not zero, see details in the Subtasks section).

For the C language the function signature is slightly different:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - `n`: the number of elements in `s` and `t` (i.e., the number of special sections),
 - the other parameters are the same as above.

Example

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

In this example there are four special sections. The best solution is to build them in the order `0,3,1,2`, and to connect them by tracks of lengths `1,2,0` respectively. This is how a train travels along this railroad track:

- Initially the speed of the train is `1` km/h.
- The train starts the ride by entering special section `0`.
- The train leaves section `0` going at `7` km/h.
- Then there is a track of length `1` m. When the train reaches the end of the track, its speed is `6` km/h.
- The train enters special section `3` going at `6` km/h and leaves it at the same speed.
- After leaving section `3`, the train travels along a `2` m long track. Its speed decreases to `4` km/h.
- The train enters special section `1` going at `4` km/h and leaves it going at `3` km/h.
- Immediately after special section `1` the train enters special section `2`.
- The train leaves section `2`. Its final speed is `8` km/h.

The function should return the total length of track between special sections:

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

Subtasks

In all subtasks $1 \leq s_i \leq 10^9$ and $1 \leq t_i \leq 10^9$.

1. (11 points): $2 \leq n \leq 8$,
2. (23 points): $2 \leq n \leq 16$,
3. (30 points): $2 \leq n \leq 200\,000$. In this subtask your program only needs to check whether the answer is zero or not. If the answer is not zero, any positive integer answer is considered correct.
4. (36 points): $2 \leq n \leq 200\,000$.

Sample grader

The sample grader reads the input in the following format:

- line 1: integer `n`.

- line $2 + i$, for i between 0 and $n - 1$: integers s_i and t_i .