

## Amerikāņu kalniņu dzelzceļš

Anna strādā atrakciju parkā un viņa atbild par dzelzceļa izbūvi jaunam amerikāņu kalniņam. Viņa jau ir izstrādājusi  $n$  speciālus *posmus* (numurētus no  $0$  līdz  $n-1$ ) kas ietekmē amerikāņu kalniņa vilciena ātrumu. Tagad viņai ir jāsaliek tos kopā un jāpiedāvā galīgais amerikāņu kalniņa dizains. Šajā uzdevumā varat uzskatīt ka vilciena garums ir nulle.

Katram  $i$  starp  $0$  un  $n-1$  ieskaitot, posmam  $i$  piemīt divas īpašības:

- iebraucot posmā vilciena ātrumam ir jābūt **mazākam vai vienādam** ar  $s_i$  km/h (kilometri stundā),
- izbraucot no posma, vilciena ātrums ir **precīzi**  $t_i$  km/h, neatkarīgi no ātruma ar kādu vilciens iebrauca posmā.

Pabeigts Amerikāņu kalniņš ir viens dzelzceļš kas satur  $n$  posmus noteiktā secībā. Katram no  $n$  posmiem ir jābūt izmantotam tieši vienreiz. Secīgus posmus savieno ar vienkāršu *trašu* palīdzību. Annai ir jāizvēlas  $n$  posmu secība un tad ir jāizlemj par tos savienojošo trašu garumiem. Katras trases garums tiek mērīts metros un var būt jebkurš nenegatīvs vesels skaitlis (iespējams, nulle).

Katrs trases metrs starp diviem posmiem samazina vilciena ātrumu par  $1$  km/h. Brauciena sākumā vilciens iebrauc amerikāņu kalniņā ar ātrumu  $1$  km/h.

Galīgam dizainam ir jāatbilst šādām prasībām:

- vilciens nepārkāpj ātruma ierobežojumu iebraucot posmos;
- vilciena ātrums jebkurā momentā līdz pēdējā speciālā posma beigū sasniegšanai ir pozitīvs.

Visos apakšuzdevumos, izņemot trešo, jums ir jāatrod minimālais iespējamais visu savienojošo trašu kopējais garums. Trešajā apakšuzdevumā jums ir tikai jāparbauda vai eksistē derīgs amerikāņu kalniņu dizains, kur katrai trasei ir nulles garums.

### Implementācijas detaļas

Jums ir jāimplementē viena funkcija (metode):

- `int64 plan_roller_coaster(int[] s, int[] t)`
  - `s`: masīvs ar garumu  $n$ , maksimālie atļautie iebraukšanas ātrumi.
  - `t`: masīvs ar garumu  $n$ , izbraukšanas ātrumi.
  - Visos apakšuzdevumos izņemot trešo, funkcijai ir jāatgriež mazākais iespējamais visu trašu starp posmiem kopējais garums. Trešajā apakšuzdevumā funkcijai ir jāatgriež  $0$ , ja eksistē derīgs amerikāņu kalniņu dizains, kur katrai trasei ir nulles garums; un jebkurš pozitīvs vesels skaitlis, ja tāds dizains neeksistē.

C valodai funkcijas signatūra ir mazliet atšķirīga:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
  - `n`: elementu skaits masīvos `s` un `t` (t.i., speciālo posmu skaits),
  - citi parametri ir tādi paši kā iepriekšējās funkcijas aprakstā.

## Piemērs

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

šajā piemērā ir četri posmi. Labākais risinājums ir izmantot tos secībā `0,3,1,2`, un savienot ar trasēm, kuru garumi ir attiecīgi `1,2,0`. Pa šo amerikāņu kalniņu vilciens brauc šādi:

- Sākotnēji, vilciena ātrums ir `1` km/h.
- Vilciens ie brauc posmā `0`.
- Vilciens izbrauc no posma `0` ar ātrumu `7` km/h.
- Pēc tam ir trase ar garumu `1` m. Kad vilciens sasniedz trases beigas, tā ātrums ir `6` km/h.
- Vilciens ie brauc posmā `3` ar ātrumu `6` km/h un izbrauc no tā ar tādu pašu ātrumu.
- Pēc izbraukšanas no posma `3`, vilciens izbrauc pa `2` m garu trasi. Tā ātrums samazinās un pēc izbraukšanas pa trasi ir `4` km/h.
- Vilciens ie brauc posmā `1` ar ātrumu `4` km/h un izbrauc no tā ar ātrumu `3` km/h.
- Uzreiz pēc posma `1` vilciens ie brauc posmā `2`.
- Vilciens izbrauc no posma `2`. Tā beigu ātrums ir `8` km/h.

Funkcijai ir jāatgriež visu trašu starp sekcijām kopējais garums:  $1 + 2 + 0 = 3$ .

## Apakšuzdevumi

Visos apakšuzdevumos  $1 \leq s_i \leq 10^9$  un  $1 \leq t_i \leq 10^9$ .

1. (11 punkti):  $2 \leq n \leq 8$ ,
2. (23 punkti):  $2 \leq n \leq 16$ ,
3. (30 punkti):  $2 \leq n \leq 200\,000$ . Šajā apakšuzdevumā jūsu programmai ir tikai jāparbauda vai atbilde ir nulle vai nē. Ja atbilde nav nulle, jebkurš pozitīvs vesels skaitlis tiks uzskatīts par pareizu.
4. (36 punkti):  $2 \leq n \leq 200\,000$ .

## Piemēru vērtētājs

Piemēru vērtētājs lasa ievaddatus šādā formātā:

- `1`. rinda: vesels skaitlis `n`.
- `2+i`. rinda, katram `i` starp `0` un `n-1`: veseli skaitļi `si` un `ti`.