

Roller Coaster Railroad

Anna werkt in een pretpark en is verantwoordelijk voor de bouw van het spoor van een nieuwe achtbaan. Zij heeft al n hindernissen ontworpen (genummerd van 0 tot en met $n - 1$) die de snelheid van de trein op de achtbaan beïnvloeden. Zij combineert de hindernissen om tot een definitief ontwerp te komen. Bij deze opgave mag je aannemen dat de trein lengte nul heeft.

Voor elke i van 0 tot en met $n - 1$ hebben de hindernissen i twee eigenschappen:

- wanneer een trein een hindernis binnenkomt geldt een maximum snelheid: de snelheid van de trein moet **gelijk of minder** zijn dan s_i km/h,
- wanneer een trein een hindernis verlaat, is de snelheid **precies** t_i km/h, ongeacht met welke snelheid de trein de hindernis binnenkwam.

De achtbaan is een lange baan van rails waarbij alle n hindernissen precies eenmaal worden aangedaan. Opvolgende hindernissen worden verbonden middels rails. Anna moet de volgorde van de n hindernissen bepalen en bepaalt ook de lengte van de stukken rails ertussen. De lengte van de stukken rails is in meters en is een niet-negatieve integer (mogelijk nul).

Elke meter rails tussen twee hindernissen verlaagt de snelheid van de trein met 1 km/h.

Aan het begin van de rit rijdt de trein de eerste van de hindernissen, in de door Anna bepaalde volgorde, binnen met een snelheid van 1 km/h.

Het uiteindelijke ontwerp moet aan de volgende voorwaarden voldoen:

- de trein mag nergens te hard een hindernis binnenkomen;
- de snelheid van de trein moet gedurende de hele rit positief blijven.

In alle subtaken, behalve voor subtaak 3, moet je de totale lengte van de stukken rails tussen de hindernissen minimaliseren. In subtaak 3 hoef je alleen te bepalen of het mogelijk is om een achtbaan te ontwerpen waarbij alle stukken rails lengte nul hebben.

Implementatie details

Je moet de volgende functie implementeren:

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : array met lengte n , de maximale toegestane snelheden.
 - t : array met lengte n , de uitgangssnelheden.
 - In alle subtasks, met uitzondering van subtask 3, moet de functie de

minimale totale lengte van alle stukken rails retourneren. In subtask 3 moet de function `0` teruggeven als er een geldige achtbaan bestaat waarbij ieder stuk rails de lengte `0` heeft, en een willekeurige positieve integer als dat niet het geval is.

Voor de taal C is de specificatie iets anders:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - `n`: het aantal elementen in `s` en `t` (het aantal hindernissen),
 - de rest is zoals hierboven vermeld.

Voorbeeld

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

In dit voorbeeld zijn er vier hindernissen. De beste oplossing is om ze in de volgorde `0,3,1,2` te plaatsen en ze te verbinden met stukken rails met lengtes `1,2,0`. Dit is hoe een trein de reis over de achtbaan maakt:

- Eerst gaat de trein met `1` km/h.
- De trein begint de rit met het binnengaan van hindernis `0`.
- De trein verlaat hindernis `0` met `7` km/h.
- Dan komt er een stuk rails van `1` m. Wanneer de trein bij het einde komt is de snelheid `6` km/h.
- De trein komt dan hindernis `3` binnen met `6` km/h en verlaat die met dezelfde snelheid.
- Na hindernis `3` gaat de trein over `2` m rails. De snelheid vermindert tot `4` km/h.
- De trein komt hindernis `1` binnen met `4` km/h en verlaat die met `3` km/h.
- Direct na het verlaten van hindernis `1` gaat de trein naar hindernis `2`.
- De trein verlaat hindernis `2` met eindsnelheid `8` km/h.

De functie retourneert de totale lengte van de rails: $1 + 2 + 0 = 3$.

Subtaken

Voor alle subtaken geldt $1 \leq s_i \leq 10^9$ en $1 \leq t_i \leq 10^9$.

1. (11 punten): $2 \leq n \leq 8$,
2. (23 punten): $2 \leq n \leq 16$,
3. (30 punten): $2 \leq n \leq 200\,000$. In deze subtaak moet je programma bepalen of het antwoord nul is of niet. Als het antwoord niet nul is, wordt iedere positieve integer goedgekeurd.
4. (36 punten): $2 \leq n \leq 200\,000$.

Voorbeeld grader

De voorbeeld grader leest de invoer als volgt in:

- regel 1: integer `n`.

- regel $2 + i$, voor i van 0 tot en met $n - 1$: integers s_i en t_i .