



Berg-og-dal-bane

Anna jobber i en fornøylespark og er ansvarlig for å bygge skinnene til en ny berg-og-dal-bane. Hun har allerede designet n spesielle seksjoner (praktisk nok nummerert fra 0 til $n - 1$) som påvirker hastigheten til toget: bakker, bremsestrek, og mange andre. Nå må hun putte de sammen og lage et forslag til design av berg-og-dal-banen. For denne oppgaven så kan du anta at lengden på toget er 0 .

For hver i fra 0 til $n - 1$ (inkludert) har den spesielle seksjonen i to egenskaper:

- det er en fartsbegrensning for å kjøre inn i seksjonen: toget må ha en fart på **maksimalt** s_i km/t når den kjører inn i seksjonen.
- når toget forlater seksjonen så er hastigheten til toget **nøyaktig** t_i km/t, uavhengig av hvilken hastighet toget hadde da det kjørte inn i seksjonen.

Den endelige berg-og-dal-banen er en bane som forbinder de n spesielle seksjonene i en eller annen rekkefølge. Hver av de n seksjonene skal brukes nøyaktig én gang. Etterfølgende seksjoner er forbundet med *spor*. Anna skal velge rekkefølgen på de n seksjonene og deretter avgjøre lengden på sporene. Lengden av et spor er gitt i meter, og kan være et hvilket som helst ikke-negativt heltall (inkludert 0).

Hver meter med spor mellom to spesielle seksjoner senker togets hastighet med 1 km/t. I begynnelsen av berg-og-dal-banen så vil toget kjøre inn i den første seksjonen (ut i fra rekkefølgen valgt av Anna) i en hastighet på 1 km/t

Det endelige designet av berg-og-dal-banen må tilfredsstillende følgende krav:

- toget må ikke overskride noen av fartsbegrensningene for å kjøre inn i de spesielle seksjonene;
- hastigheten av toget må være positive til enhver tid.

I alle subtasks untatt i subtask 3 så er oppgaven din å finne den minste mulige totale lengde med spor som må legges til for å lage en gyldig berg-og-dal-bane. I subtask 3 så trenger du bare å sjekke om det finnes et gyldig design hvor alle sporene har lengde 0 .

Implementasjonsdetaljer

Du skal implementere følgende funksjon:

- `int64 plan_roller_coaster(int[] s, int[] t)`
 - s : et array av lengde n , de største tillatte hastighetene når toget kjører inn på seksjonene

- t : et array av lengde n , hastighetene toget har når det forlater seksjonene.
- I alle subtasks untatt subtask 3 så skal funksjonen returnere den minste mulige totale lengde av alle spor. I subtask 3 skal funksjonen returnere 0 dersom det finnes et gyldig design hvor hvert spor har lengde 0, og et hvilket som helst positivt heltall dersom en slik bane ikke finnes.

IC er funksjonen litt annerledes:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - n : antall elementer i s og t (dvs. antall spesielle seksjoner),
 - de andre parameterene er som over

Eksempel

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

Her er det fire spesielle seksjoner. Den beste løsningen er å bygge de i rekkefølge $0, 3, 1, 2$, og å forbinde de med spor med lengder $1, 2, 0$ respektivt. Dette er hvordan et tog reiser langs sporet:

- Først har toget en hastighet på 1 km/t.
- Toget begynner berg-og-dal-banen med å kjøre inn i seksjon 0 .
- Toget forlater seksjon 0 i en hastighet på 7 km/t.
- Deretter er det et spor med lengde 1 m. Når toget kommer til enden av sporet har det en hastighet på 6 km/t.
- Toget kjører inn i seksjon 3 i 6 km/t og forlater den i samme fart.
- Etter å ha forlatt seksjon 3 reiser toget langs 2 m med bane, som senker farten dens til 4 km/t.
- Toget kjører inn i seksjon 1 i 4 km/t og forlater den i 3 km/t.
- Umiddelbart etter seksjon 1 kjører toget inn på seksjon 2 .
- Toget forlater seksjon 2 . Dens slutthastighet er 8 km/t.

Funksjonen skal returnere den totale lengden av sporene mellom seksjonene:

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

Subtasks

I alle subtasks: $1 \leq s_i \leq 10^9$ og $1 \leq t_i \leq 10^9$.

1. (11 poeng): $2 \leq n \leq 8$,
2. (23 poeng): $2 \leq n \leq 16$,
3. (30 poeng): $2 \leq n \leq 200\,000$. Her trenger programmet ditt bare å sjekke om svaret er 0 eller ikke. Dersom svaret ikke er 0 så vil alle positive heltall regnes som riktig.
4. (36 points): $2 \leq n \leq 200\,000$.

Sample grader

Graderen leser input i følgende format:

- linje 1: heltall n .
- linje $2 + i$, for i mellom 0 og $n - 1$: heltall s_i og t_i .