

Roller Coaster Railroad

Anna lucrează la un parc de distracții, iar acum coordonează construcția unui nou roller-coaster. Ea a proiectat deja n sectoare speciale (indexate convenabil de la 0 la $n - 1$) care afectează viteza trenului: sectoare ascendente, sectoare de frânare și altele. Acum trebuie să assembleze toate aceste sectoare și să propună o versiune finală a roller-coaster-ului. În această problemă, vom considera că trenul este punctiform (are lungime egală cu 0).

Pentru fiecare i între 0 și $n - 1$ inclusiv, cel de al i -lea sector special are două proprietăți:

- atunci când trenul intră în acest sector, viteza sa trebuie să fie mai mică sau egală cu s_i km/h.
- atunci când părăsește acest sector, viteza trenului va fi **exact** t_i km/h, indiferent de viteza cu care trenul a intrat în acest sector.

Roller-coaster-ul va fi asamblat din aceste n sectoare speciale, care vor fi așezate într-o anumită ordine. Fiecare din cele n sectoare speciale trebuie să apară exact o dată în roller-coaster. Mai mult, trebuie să existe un segment de cale ferată între oricare două sectoare consecutive. Anna trebuie să stabilească ordinea celor n sectoare, iar apoi să decidă lungimea fiecărui segment de cale ferată. Lungimea unui segment este măsurată în metri și poate fi egală cu orice număr natural nenegativ (inclusiv 0).

Fiecare metru de cale ferată dintre două sectoare speciale încetinește trenul cu 1 km/h. La începutul călătoriei trenul intră în primul sector special (în ordinea aleasă de Anna) cu viteza de 1 km/h.

Versiunea finală a roller-coaster-ului trebuie să respecte următoarele cerințe:

- trenul nu încalcă nicio limită de viteză atunci când intră în sectoarele speciale.
- viteza trenului rămâne permanent pozitivă.

În toate subtask-urile, cu excepția subtaskului 3, sarcina voastră este să găsiți o ordonare a celor n sectoare speciale și să decideți lungimile segmentelor de cale ferate dintre sectoarele consecutive, astfel încât lungimea totală a segmentelor de cale ferată să fie minimă. În subtask-ul 3 trebuie doar să verificați dacă există o ordonare a sectoarelor speciale astfel încât această lungime totală să fie egală cu zero.

Detalii de Implementare

Trebuie să implementați următoarea funcție (metodă):

- `int64 plan_roller_coaster(int[] s, int[] t)`

- **s**: șir de lungime n , vitezele maxime de intrare.
- **t**: șir de lungime n , vitezele de ieșire.
- În fiecare subtask, cu excepția subtaskului 3, funcția trebuie să întoarcă lungimea totală minimă posibilă a segmentelor de cale ferată. În subtaskul 3 funcția va returna 0 dacă există un roller coaster design valid, astfel încât fiecare segment are lungime nulă, sau oricare număr pozitiv întreg dacă un asemenea design nu există.

Pentru limbajul C, prototipul funcției este ușor diferit:

- `int64 plan_roller_coaster(int n, int[] s, int[] t)`
 - **n**: numărul de elemente din **s** și **t** (i.e., numărul de sectoare speciale),
 - ceilalți parametri sunt identici cu cei precizați mai sus.

Exemple

Exemplul 1

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

În acest exemplu există patru sectoare speciale. Cea mai bună soluție este să le construim în ordinea 0, 3, 1, 2 și să le conectăm cu segmente de lungime 1, 2, 0. Trenul va călători astfel:

- Inițial, viteza trenului este de 1 km/h.
- Trenul începe cursa intrând în sectorul special cu numărul 0.
- Trenul părăsește sectorul 0 cu o viteză de 7 km/h.
- Urmează apoi un segment de lungime egală cu 1 metru. Când trenul ajunge la finalul acestui segment, viteza sa este 6 km/h.
- Trenul intră în sectorul cu numărul 3 cu o viteză de 6 km/h și îl părăsește cu aceeași viteză.
- După ce părăsește sectorul 3, trenul călătorește de-a lungul unui segment de 2 metri. Viteza sa scade la 4 km/h.
- Trenul intră apoi în sectorul 1 cu o viteză de 4 km/h și îl părăsește cu o viteză de 3 km/h.
- Imediat după sectorul special 1, trenul intră în sectorul special cu numărul 2.
- Trenul părăsește sectorul cu numărul 2. Viteza sa finală este de 8 km/h.

Funcția trebuie să întoarcă lungimea totală a segmentelor de cale ferată: $1 + 2 + 0 = 3$.

Subtaskuri

În toate subtask-urile $1 \leq s_i \leq 10^9$ și $1 \leq t_i \leq 10^9$.

1. (11 puncte): $2 \leq n \leq 8$,
2. (23 de puncte): $2 \leq n \leq 16$,
3. (30 de puncte): $2 \leq n \leq 200\,000$. În acest subtask programul vostru trebuie doar să verifice dacă răspunsul este egal cu 0. Dacă răspunsul nu este 0, orice număr natural pozitiv este considerat corect.
4. (36 de puncte): $2 \leq n \leq 200\,000$.

Sample grader

Sample grader-ul citește datele în următorul format:

- linia 1: număr întreg n .
- linia $2 + i$, pentru i între 0 și $n - 1$: numerele întregi s_i și t_i .