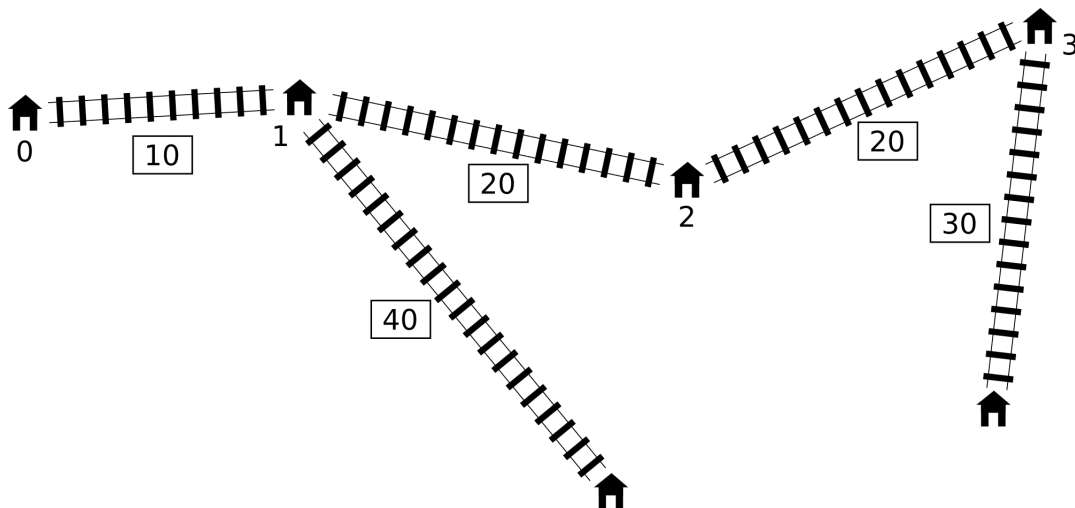


Oikorata

Kaalepilla on lelujunarata, joka on hyvin yksinkertainen. Siinä on yksi päärata, jonka varrella on n asemaa. Asemat on numeroitu $0 \dots n - 1$ niiden järjestyksessä pääradalla. Asemien i ja $i + 1$ välinen etäisyys on l_i cm ($0 \leq i < n - 1$).

Pääradan lisäksi radassa voi olla sivuraiteita. Jokainen sivuraide lähtee tietyistä pääradan asemasta ja päättyy sivuasemaan, joka ei ole pääradalla. (Sivuasemia ei ole numeroitu.) Jokaisesta pääradan asemasta voi alkaa enintään yksi sivuraide. Asemasta i alkavan sivuraiteen pituus on d_i cm. Merkitsemme $d_i = 0$, jos asemasta i ei lähde sivuraidetta.



Kaaleppi aikoo nyt rakentaa yhden oikoradan, joka tulee kulkemaan kahden **pääradan** aseman välillä. Asemat voivat olla vierekkäiset. Oikoradan pituus tulee olemaan tarkalleen c cm riippumatta siitä, mitkä asemat se yhdistää.

Jokaista rataosuutta, mukaan lukien uutta oikorataa, voi kulkea kumpaankin suuntaan. *Välimatka* kahden aseman välillä on pienin reitin pituus, joka yhdistää asemat raiteita pitkin. Koko rataverkon *läpimitta* on suurin välimatka kahden aseman välillä. Toisin sanoen tämä on pienin luku t niin, että minkä tahansa kahden aseman välinen etäisyys on enintään t .

Kaaleppi haluaa rakentaa oikoradan niin, että lopullisen junaverkon läpimitta on pienin mahdollinen.

Toteutuksen yksityiskohdat

Toteuta funktio

```
int64 find_shortcut(int n, int[] l, int[] d, int c)
```

- o n : asemien määrä pääradalla,
- o l : pääradan asemien etäisyydet (taulukko kokoa $n - 1$),
- o d : sivuraiteiden pituudet (taulukko kokoa n),
- o c : uuden oikoradan pituus.
- o funktion tulee palauttaa pienin mahdollinen junaverkon läpimitta oikoradan lisäämisen jälkeen.

Annetuista koodipohjista selviää, miten toteutus tulee tehdä tarkalleen.

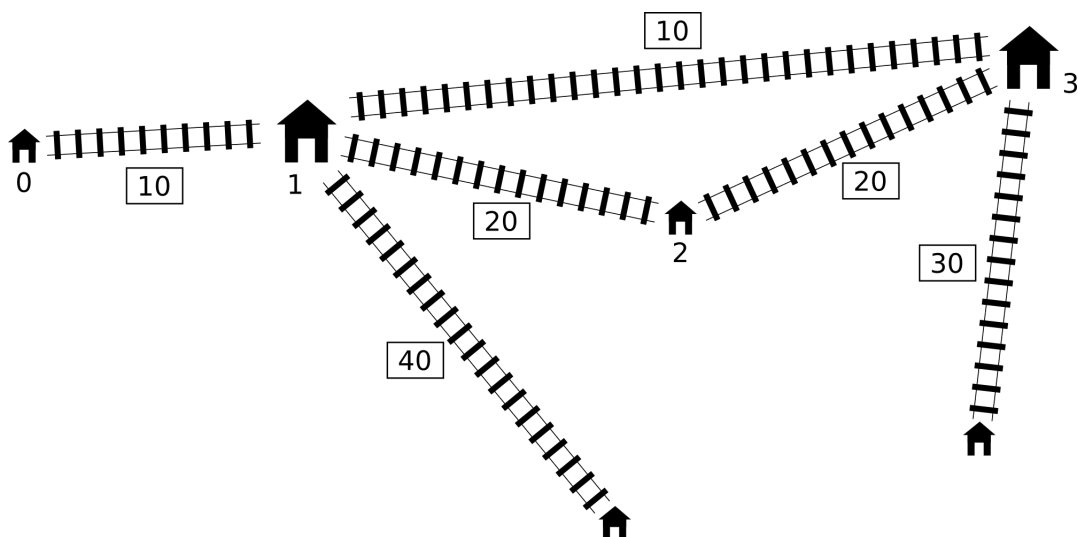
Esimerkit

Esimerkki 1

Yllä olevassa rataverkossa arvostelija tekee seuraavan kutsun:

```
find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)
```

Optimiratkaisu on rakentaa oikorata asemien 1 ja 3 välille alla olevalla tavalla.



Uuden rataverkon läpimitta on **80** cm, joten funktion tulee palauttaa **80**.

Esimerkki 2

Arvostelija tekee seuraavan funktiokutsun:

```
find_shortcut(9, [10, 10, 10, 10, 10, 10, 10, 10],  
[20, 0, 30, 0, 0, 40, 0, 40, 0], 30)
```

Optimiratkaisu on yhdistää asemat **2** ja **7**, jolloin läpimitta on **110**.

Esimerkki 3

Arvostelija tekee seuraavan funktiokutsun:

```
find_shortcut(4, [2, 2, 2],
```

```
[1, 10, 10, 1], 1)
```

Optimiratkaisu on yhdistää asemat 1 ja 2, jolloin läpimitta on 21.

Esimerkki 4

Arvostelija tekee seuraavan funktiokutsun:

```
find_shortcut(3, [1, 1],  
              [1, 1, 1], 3)
```

Minkä tahansa asemaparin yhdistäminen 3-pituaisella oikoradalla ei paranna rataverkon alkuperäistä läpimittaa, joka on 4.

Osatehtävät

Kaikissa osatehtävissä $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

1. (9 pistettä) $2 \leq n \leq 10$,
2. (14 pistettä) $2 \leq n \leq 100$,
3. (8 pistettä) $2 \leq n \leq 250$,
4. (7 pistettä) $2 \leq n \leq 500$,
5. (33 pistettä) $2 \leq n \leq 3000$,
6. (22 pistettä) $2 \leq n \leq 100\,000$,
7. (4 pistettä) $2 \leq n \leq 300\,000$,
8. (3 pistettä) $2 \leq n \leq 1\,000\,000$.

Esimerkkiarvostelija

Esimerkkiarvostelija lukee syötteen seuraavassa muodossa:

- o rivi 1: kokonaisluvut n ja c ,
- o rivi 2: kokonaisluvut l_0, l_1, \dots, l_{n-2} ,
- o rivi 3: kokonaisluvut d_0, d_1, \dots, d_{n-1} .