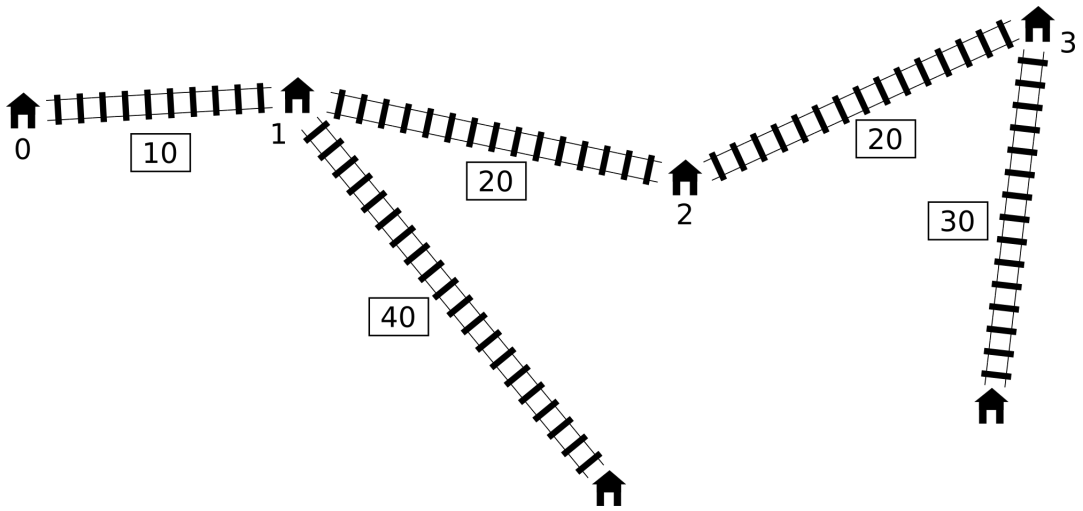
Shortcut

Pavle se voli igrati vozićima. Kupio je jednostavnu prugu koja se sastoji od n stanica koje su redom označene brojevima od 0 do $n-1$. Stanice 0 i $n-1$ leže na krajevima glavne pruge. Udaljenost između stanica i i $i+1$ iznosi l_i centimetara ($0 \leq i < n-1$).

Osim glavne pruge mogu postojati i sporedne pruge. Svaka sporedna pruga povezuje stanicu na glavnoj pruzi i novu stanicu koja ne leži na glavnoj pruzi. (Ove nove stanice nisu numerisane.) Najviše jedna sporedna pruga može početi na svakoj stanici glavne pruge. Dužina sporedne pruge koja počinje na stanici i iznosi d_i centimetara. Ako je $d_i = 0$, to znači da na stanici i nema sporedne pruge.



Pavle želi izgraditi prečicu: brzu prugu između dvije (možda i susjedne) stanice **glavne pruge**. Brza pruga biće dugačka tačno c centimetara, bez obzira koje će dvije stanice povezivati.

Sve su pruge dvosmjerne. *Udaljenost* dvije stanice je najmanja dužina dionice koja prugama ide od jedne do druge stanice. *Dijametar* cijele mreže je maksimalna udaljenost među svim parovima stanica. Drugim riječima, to je najmanji broj t takav da je udaljenost svake dvije stanice najviše t.

Pavle želi izgraditi brzu prugu tako da minimizira dijametar dobijene mreže.

Detalji implementacije

Potrebno je da implementirate funkciju (metod): `int64 find_shortcut(int n, int[] l, int[] d, int c)`

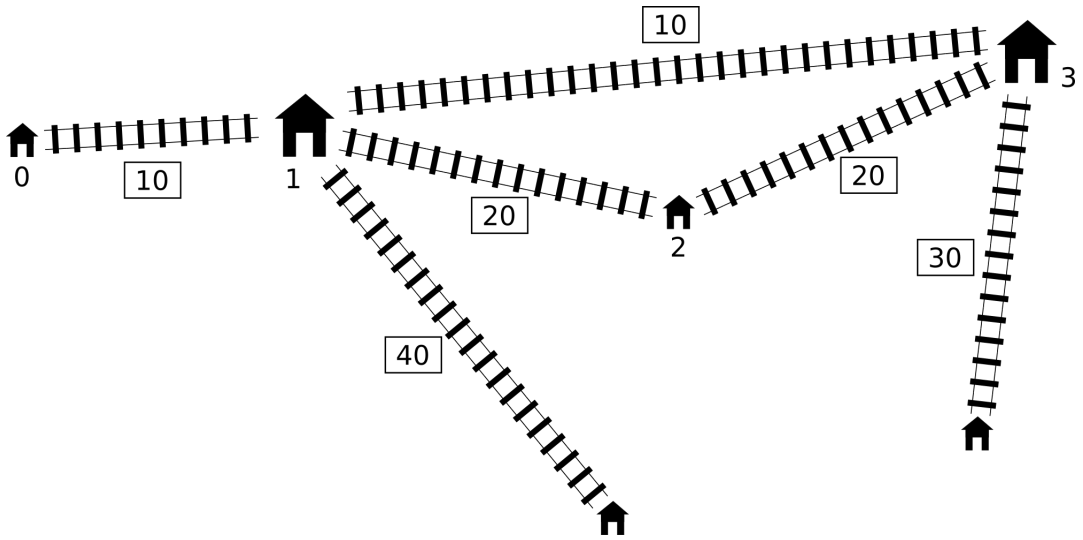
- `n`: broj stanica na glavnoj pruzi,
- `l`: udaljenosti među uzastopnim stanicama na glavnoj pruzi (niz dužine $n-1$),
- `d`: dužine sporednih pruga (niz dužine n),
- `c`: dužina nove brze pruge.
- Funkcija vraća najmanji mogući dijametar mreže nakon dodavanja brze pruge.

Molimo vas da koristite date templejt-fajlove za odgovarajući programski jezik.

Primjeri

Primjer 1

Za mrežu sa gornje slike program za ocjenjivanje će pozvati:
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`
Optimalno je rješenje izgraditi brzu prugu između stanica 1 i 3, kao na slici ispod.



Dijametar nove mreže iznosi 80 centimetara, tj. funkcija treba vratiti 80.

Primjer 2

Program za ocjenjivanje poziva:

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

Optimalno rješenje povezuje stanice 2 i 7, a dijametar u tom slučaju iznosi 110.

Primjer 3

Program za ocjenjivanje poziva:

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

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

Optimalno rješenje povezuje stanice 1 i 2, smanjujući dijametar na 21.

Primjer 4

Program za ocjenjivanje poziva:

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

Povezivanje bilo kojih dviju stanica brzom prugom dužine 3 ne poboljšava početni dijametar mreže koji iznosi 4.

Podzadaci

U svim podzadacima $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

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

Sample grader

Sistem za ocjenjivanje učitava ulazne podatke u sljedećem formatu:

- red 1: cijeli brojevi n i c,
- red 2: cijeli brojevi $l_0, l_1, \ldots, l_{n-2}$,
- red 3: cijeli brojevi $d_0, d_1, \ldots, d_{n-1}$.