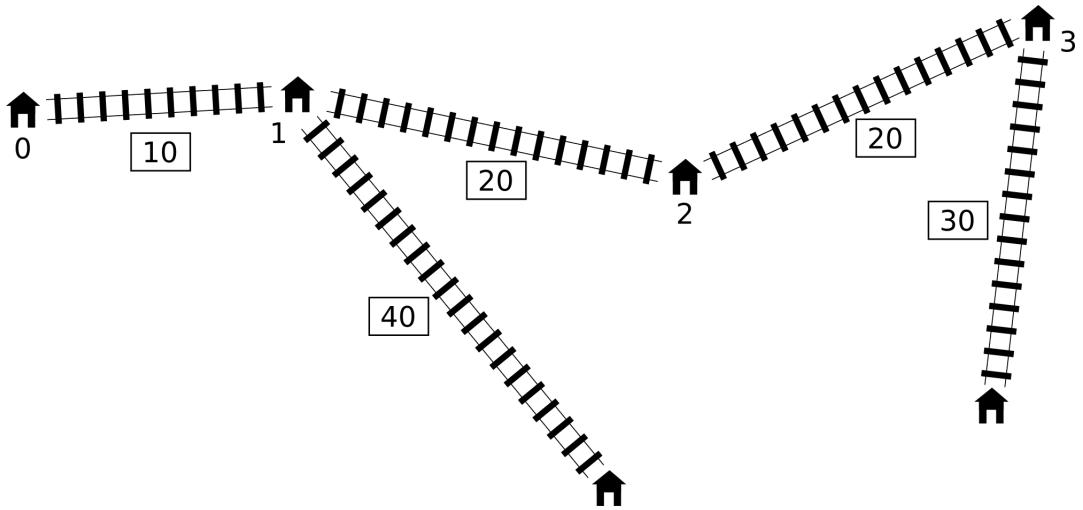
Shortcut

Nenad voli da se igra sa vozićima jer je to jednostavno i ne zahteva intelektualni napor. Kupio je jednostavnu glavnu prugu koja se sastoji od n stanica koje su označene brojevima od 0 do $n-1$, redom s leva nadesno. 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 neku novu stanicu koja nije na glavnoj pruzi. (Ove nove stanice nisu numerirane.) Iz svake stanice na glavnoj pruzi polazi najviše jedna sporedna pruga. Dužina sporedne pruge koja počinje iz stanice i iznosi d_i centimetara. Koristimo $d_i = 0$ ako iz stanice i ne polazi sporedna pruga.



Nenad voli da koristi prečice u životu pa zato i sada planira da izgradi prečicu: brzu prugu između dve različite (možda susedne) stanice sa **glavne pruge**. Brza pruga će biti dugačka tačno c centimetara, bez obzira koje će dve stanice povezivati.

Sve pruge (glavne, sporedne i brza pruga) su dvosmerne. *Udaljenost* između dve stanice je najmanja dužina rute koja prugama ide od jedne do druge stanice. *Dijametar* cele mreže pruga je maksimalna udaljenost među svim parovima stanica. Drugim rečima, to je najmanji broj t takav da je udaljenost između svake dve stanice najviše t.

Nenad želi da izgardi brzu prugu tako da minimizuje dijametar rezultujuće mreže

pruga jer tako je u mogućnosti.

Detalji implementacije

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

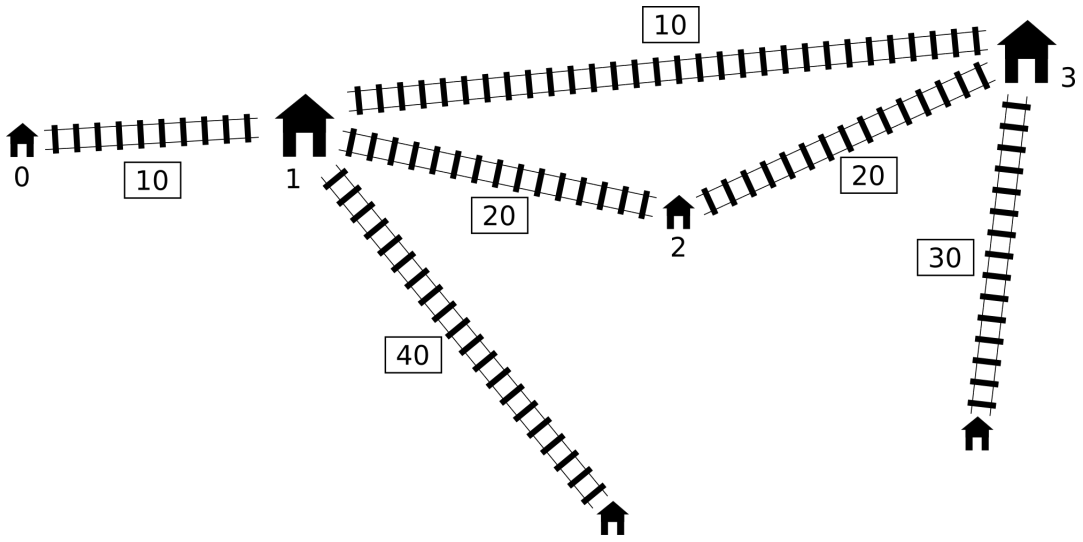
- `n`: broj stanica na glavnoj pruzi,
- `l`: udaljenosti između uzastopnih stanica na glavnoj pruzi (niz dužine $n-1$),
- `d`: dužine sporednih pruga (niz dužine n),
- `c`: dužina nove brze pruge.
- funkcija treba da vrati najmanji mogući dijametar mreže pruga nakon dodavanja brze pruge.

Koristite date templejt-fajlove za bolji uvid u detalje implementacije za vaš programski jezik.

Primeri

Primer 1

Za mrežu pruga s gornje slike grejder će pozvati:
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`
Optimalno rešenje je izgraditi brzu prugu između stanica 1 i 3, kao na slici ispod.



Dijametar nove mreže iznosi 80 centimetara, tj. funkcija treba da vrati 80.

Primer 2

Grejder poziva:

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

Optimalno rešenje je povezati stanice 2 i 7 i u tom slučaju je dijametar 110.

Primer 3

Grejder poziva:

```
find_shortcut(4, [2, 2, 2],
  [1, 10, 10, 1], 1)
```

Optimalno rešenje je povezati stanice 1 i 2 i u tom slučaju se dijametar smanjuje na 21.

Primer 4

Grejder poziva:

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

Povezivanje bilo koje dve stanice brzom prugom dužine 3 ne smanjuje početni dijametar mreže koji je 4.

Podzadaci

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

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

Opis priloženog grejdera

Priloženi grejder čita ulaz u sledećem formatu:

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