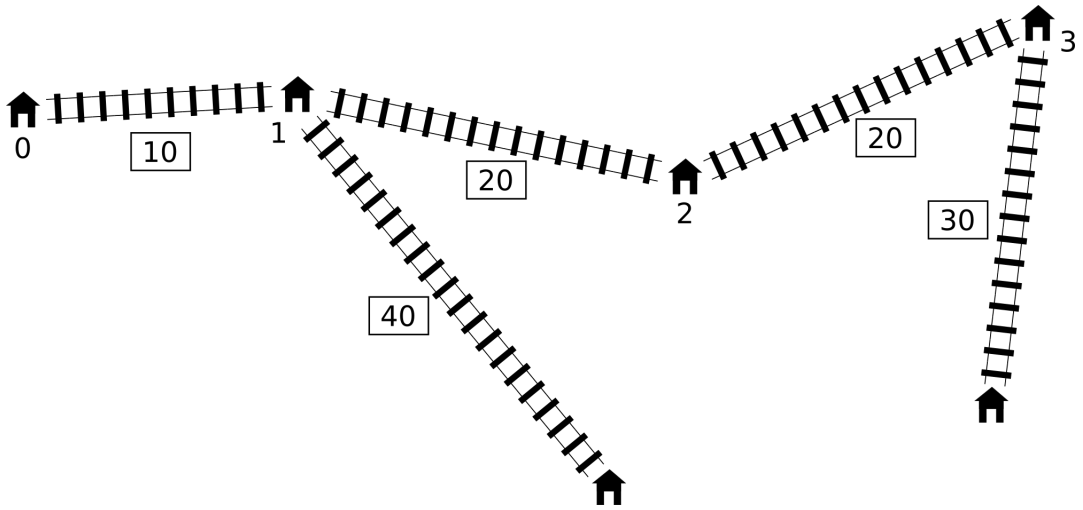
Īsceļš

Pāvelam ir vienkāršs rotaļu dzelzceļš. Tam ir viena *galvenā* līnija uz kuras atrodas n stacijas, kas tiek numurētas tādā secībā kā tās atrodas uz līnijas ar skaitļiem no 0 līdz $n-1$ pēc kārtas. Attālums starp stacijām i un $i+1$ ir l_i centimetri ($0 \leq i < n-1$).

Bez galvenās līnijas dzelzceļam var būt arī *papildus* līnijas. Katra papildus līnija ir dzelzceļš starp staciju uz galvenās līnijas un kādu jaunu staciju kura nepieder galvenai līnijai (šīs stacijas netiek numurētas). No katras stacijas uz galvenās līnijas var iziet ne vairāk kā viena papildus līnija. Papildus līnijas, kas sākas stacijā i , garums ir d_i centimetri. Mēs lietosim $d_i = 0$ lai apzīmētu to, ka neeksistē papildus līnija, kas sākas stacijā i .



Pāvels plāno uzbūvet vienu īsceļu: ekspreslīniju starp divam dažādam (iespējams arī, ka kaimiņu) stacijām uz **galvenās līnijas**. Ekspreslīnijas garums būs precīzi c centimetri, neatkarīgi no tā, kuras divas stacijas tiks savienotas.

Katrs dzelzceļa segments, ieskaitot jauno ekspreslīniju, var tikt izmantots abos virzienos. *Attālums* starp divām stacijam ir maršruta, kas pa dzelzceļa segmentiem iet no vienas pilsētas uz otru, mazākais iespējamais garums. Visa dzelzceļa tīkla *diametrs* ir maksimālais attālums starp visiem staciju pāriem. Citiem vārdiem, tas ir tāds mazākais skaitlis t , ka attālums starp stacijām katram staciju pārim nepārsniedz t .

Pāvels grib uzbūvēt ekspreslīniju tā, lai rezultējošā dzelzceļa tīkla diametrs būtu

mazākais iespējamais.

Implementācijas detaļas

Jums ir jāimplementē viena funkcija (metode):
`int64 find_shortcut(int n, int[] l, int[] d, int c)`

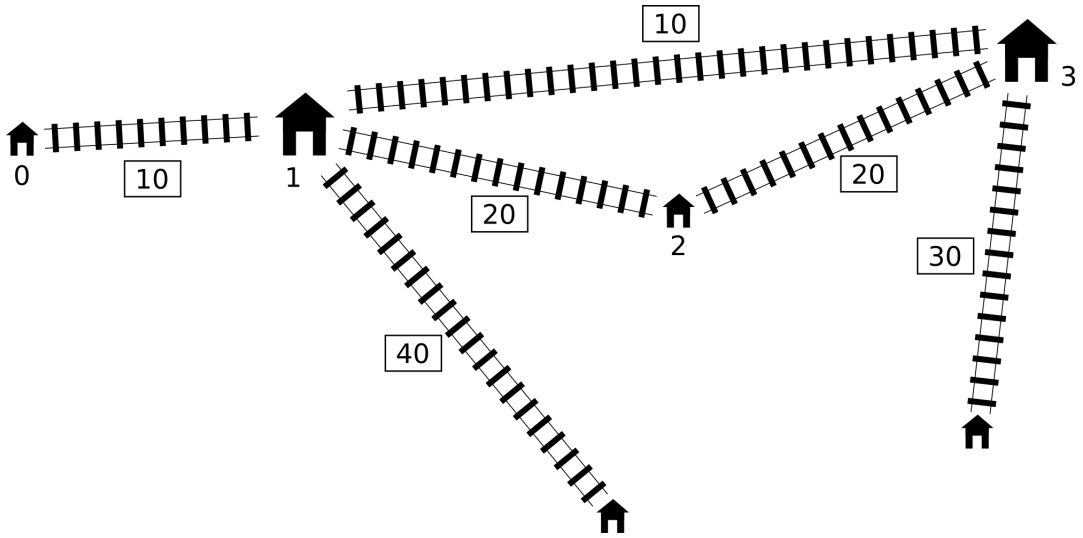
- `n`: staciju skaits uz galvenās līnijas,
- `l`: attālumi starp stacijam uz galvenās līnijas (masīvs garumā $n-1$),
- `d`: papildus līniju garumi (masīvs garumā n),
- `c`: jaunās ekspreslīnijas garums.
- funkcijai ir jāatgriež mazākais iespējamais dzelzceļa tīkla diametrs pēc ekspreslīnijas izbūves.

Implementācijas detaļām lūdzu izmantojiet piedāvāos šablona failus jūsu izmantotajā programmēšanas valodā.

Piemēri

1. piemērs

Augstāk attēlotajam dzelzceļu tīklam vērtētājs veiks šādu funkcijas izsaukumu:
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`
Optimāls atrisinājums ir uzbūvēt ekspreslīniju starp stacijam 1 un 3:



Jauna dzelzceļu tīkla diametrs ir 80 centimetri, tāpēc funkcijai ir jāatgriež 80.

2. piemērs

Vērtētājs veiks šādu funkcijas izsaukumu:

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

Optimāls atrisinājums ir savienot stacijas 2 un 7. Šajā gadījumā diametrs ir 110.

3. piemērs

Vērtētājs veiks šādu funkcijas izsaukumu:

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

Optimāls atrisinājums ir savienot stacijas 1 un 2. Šajā gadījumā diametrs samazinās līdz 21.

4. piemērs

Vērtētājs veiks šādu funkcijas izsaukumu:

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

Savienojot jebkuras divas stacijas ar ekspreslīniju ar garumu 3 neuzlabo sākotnējo dzelzceļu tīkla diametru, kas ir 4.

Apakšuzdevumi

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

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

Piemēru vērtētājs

Piemēru vērtētājs lasa ievaddatus šādā formātā:

- 1. rinda: veseli skaitļi n un c,
- 2. rinda: veseli skaitļi $l_0, l_1, \ldots, l_{n-2}$,
- 3. rinda: veseli skaitļi $d_0, d_1, \ldots, d_{n-1}$.