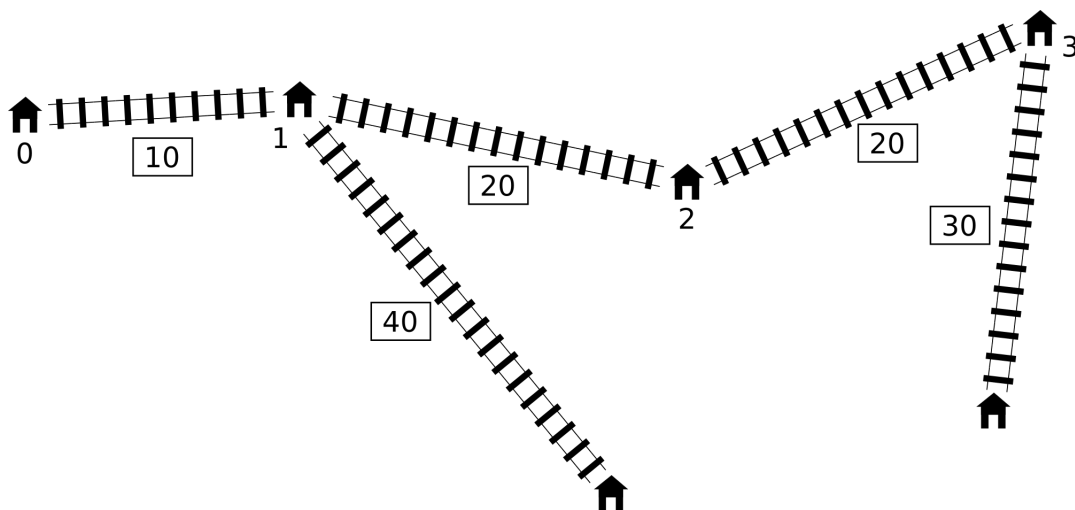


## Zkratka

Pavel má modelovou železnici. Je velmi prostá. Je zde hlavní trať o  $n$  stanicích číselovaných postupně od  $0$  do  $n - 1$ . Stanice  $0$  a  $n - 1$  leží na koncích hlavní tratě. Vzdálenost mezi stanicemi  $i$  a  $i + 1$  je  $l_i$  centimetrů ( $0 \leq i < n - 1$ ).

Vedle hlavní trati zde mohou být trati vedlejší. Každá vedlejší trať vede ze stanice na hlavní trati do nějaké nové stanice, která na hlavní trati neleží. Nové stanice všech vedlejších tratí jsou navzájem různé a nové stanice nejsou číselované. V každé stanici hlavní trati začíná nejvýše jedna trať vedlejší. Délka vedlejší tratě ze stanice  $i$  je  $d_i$  centimetrů. Pomocí  $d_i = 0$  značíme, že ze stanice  $i$  nevede žádná vedlejší trať.



Pavel plánuje zbudovat jednu zkratku: expresní trať mezi dvěma (možno i sousedními) různými stanicemi na **hlavní trati**. Expresní trať bude mít délku přesně  $c$  centimetrů bez ohledu na to, které stanice spojí.

Všechny úseky trati včetně nové expresní tratě jsou obousměrné. *Vzdáleností* mezi dvěma stanicemi je nejmenší délka cesty mezi těmito dvěma stanicemi. *Průměrem* celé železniční sítě je maximální vzdálenost mezi všemi dvojicemi stanic. Jinými slovy, je to nejmenší číslo  $t$  takové, že vzdálenost mezi každými dvěma stanicemi je nejvýše  $t$ .

Pavel chce zbudovat expresní trať tak, aby průměr výsledné železniční sítě byl co nejmenší.

### Implementační detaily

Implementujte funkci

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

- `n`: počet stanic na hlavní trati,
- `l`: vzdálenosti mezi sousedními stanicemi (pole délky  $n - 1$ ),
- `d`: délky vedlejších tratí (pole délky  $n$ ),
- `c`: délka nové expresní tratě.
- funkce vrací nejmenší možný průměr železniční sítě po přidání expresní tratě.

V šablonách souborů najdete další detaily pro implementaci ve zvoleném programovacím jazyce.

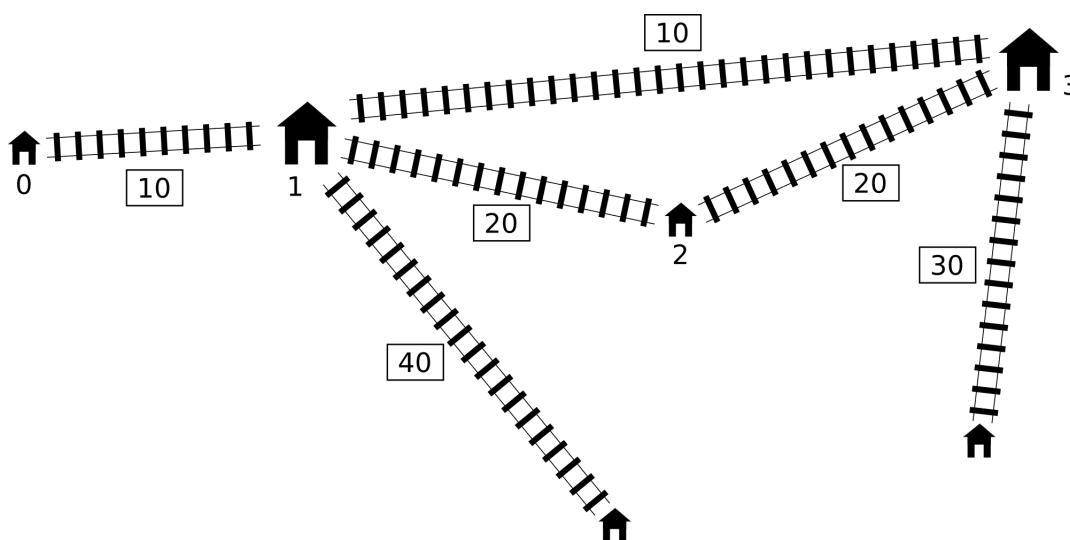
## Příklady

### Příklad 1

Pro železniční síť uvedenou výše provede vyhodnocovač následující volání funkce:

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

Optimálním řešením je spojit expresní tratí stanice 1 a 3, viz obrázek.



Průměr nové železniční sítě je pak **80** centimetrů, takže funkce má vrátit hodnotu **80**.

### Příklad 2

Vyhodnocovač provede následující volání funkce:

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

Optimálním řešením je spojit expresní tratí stanice **2** a **7**. V tomto případě je průměr výsledné železniční sítě **110** centimetrů.

### Příklad 3

Vyhodnocovač provede následující volání funkce:

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

Optimálním řešením je spojit expresní tratí stanice **1** a **2**, což sníží průměr na **21**.

#### Příklad 4

Vyhodnocovač provede následující volání funkce:

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

Žádné spojení dvou stanic expresní tratí délky **3** nezlepší výchozí průměr sítě **4** centimetry.

#### Podúlohy

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

#### Vzorový vyhodnocovač

Vzorový vyhodnocovač čte vstup v následujícím formátu:

- řádek 1: celá čísla  $n$  a  $c$ ,
- řádek 2: celá čísla  $l_0, l_1, \dots, l_{n-2}$ ,
- řádek 3: celá čísla  $d_0, d_1, \dots, d_{n-1}$ .