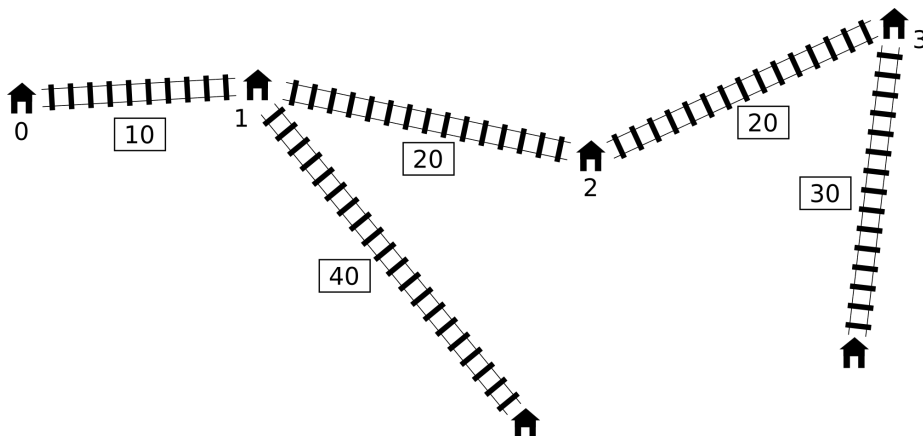


## Շրջանցող ճանապարհ

Պաժելը ունի խաղալիք-երկաթուղի: Այն շատ պարզ է: Երկաթուղին ունի ճիշտ մեկ գլխավոր գիծ, որը իրար է միացնում  $n$  կայարաններ՝ հաջորդաբար համարակալված  $0$ -ից  $n - 1$  թվերով:  $i$  և  $i + 1$  կայարանների միջև հեռավորությունը կազմում է  $l_i$  սմ ( $0 \leq i < n - 1$ ):

Բացի գլխավոր գծից կարող են գոյություն ունենալ նաև երկրորդային գծեր: Երկրորդային գիծը իրենից ներկայացնում է մի գիծ, որը միացնում է գլխավոր գծի վրա գտնվող որևէ կայարան մեկ այլ՝ գլխավոր գծի վրա չգտնվող կայարանի հետ (վերջիններս համարակալված չեն): Գլխավոր գծի վրա գտնվող ամեն մի կայարանից կարող է դուրս գալ ամենաշատը մեկ երկրորդային գիծ:  $i$  կայարանից սկսվող երկրորդային գծի երկարությունը կազմում է  $d_i$  սմ: Կասենք, որ  $d_i = 0$  եթե  $i$  կայարանից ելնող երկրորդային գիծ գոյություն չունի:



Պաժելը որոշել է ավելացնել մեկ շրջանցող ճանապարհ՝ գիծ, որը պետք է միացնի **գլխավոր գծի** վրա գտնվող երկու տարբեր կայարան (որոնք կարող են լինել իրար հարևան): Շրջանցող ճանապարհի երկարությունը կազմում է ճիշտ  $c$  սմ, անկախ նրանից թե որ կայարաններն է միացնում իրար:

Բոլոր ուղիները, ներառյալ շրջանցող գիծը, երկկողմանի են: Երկու կայարանների միջև հեռավորությունը սահմանվում է որպես նրանց միացնող ճանապարհների երկարություններից ամենափոքրը: Երկաթուղու տրամագիծ կանվանենք կայարանների բոլոր զույգերի միջև հեռավորություններից ամենամեծի արժեքը: Այլ կերպ ասած, այն ամենափոքր  $t$  թիվը, որ ցանկացած կայարանների զույգի միջև հեռավորությունը չի գերազանցում  $t$ -ն:

Պաժելը ցանկանում է կառուցել շրջանցող ճանապարհը այնպես, որ արդյունքում երկաթուղու տրամագիծը լինի որքան հնարավոր է փոքր:

## Իրականացման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ֆունկցիան

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

- **n**: գլխավոր գծի կայարանների քանակը,
- **l**: գլխավոր գծի կայարանների միջև հեռավորությունները ( $n - 1$  երկարությամբ զանգված),
- **d**: երկրորդային գծերի հեռավորությունները ( $n$  երկարությամբ զանգված),
- **c**: շրջանցող ճանապարհի երկարությունը:
- ֆունկցիան պետք է վերադարձնի հնարավոր ամենափոքր տրամագիծը՝ երկաթգծին շրջանցող ճանապարհ ավելացնելուց հետո:

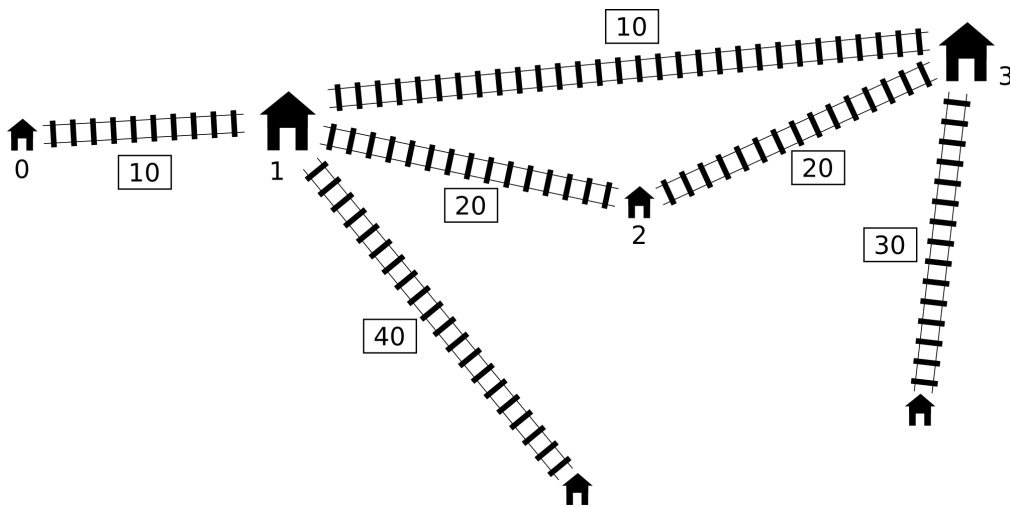
Ֆունկցիայի տեսքի մասին ավելի ճշգրիտ տեղեկություններ ստանալու համար խնդրում ենք անպայման ծանոթանալ ձեր լեզվի համար տրամադրված սմուլշային ֆայլերին:

## Օրինակներ

### Օրինակ 1

Վերը նշված օրինակի համար գրեյդերը կատարում է հետևյալ ֆունկցիայի կանչը՝  
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`

Այս դեպքում օպտիմալ լուծումը **1** և **3** համարով կայարանները միացնելն է:



Նոր երկաթուղու տրամագիծը կլինի **80** սմ, այնպես որ ֆունկցիան պետք է վերադարձնի 80:

### Օրինակ 2

Գրեյդերը կատարում է հետևյալ ֆունկցիայի կանչը՝

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

Օպտիմալ լուծումը **2** և **7** համարներով կայարանները միացնելն է: Արդյունքում տրամագիծը կստացվի **110** սմ:

### Օրինակ 3

Գրեյդերը կատարում է հետևյալ ֆունկցիայի կանչը՝

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

Օպտիմալ լուծումը 1 և 2 համարներով կայարանները միացնելն է: Արդյունքում տրամագիծը կստացվի 21 սմ:

### Օրինակ 4

Գրեյդերը կատարում է հետևյալ ֆունկցիայի կանչը՝

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

Կայարանների ոչ մի զույգ 3 երկարությամբ շրջանցող ճանապարհով միացնելու դեպքում տրամագիծը չի նվազի, որը սկզբում կազմում էր 4 :

### Ենթախնդիրներ

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

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

### Գրեյդերի օրինակ

Գրեյդերին մուտքային տվյալները տրամադրվում են հետևյալ ֆորմատով՝

- տող 1: երկու ամբողջ թվեր՝  $n$  և  $c$ ,
- տող 2:  $n - 1$  հատ ամբողջ թվեր՝  $l_0, l_1, \dots, l_{n-2}$ ,
- տող 3:  $n$  հատ ամբողջ թվեր՝  $d_0, d_1, \dots, d_{n-1}$