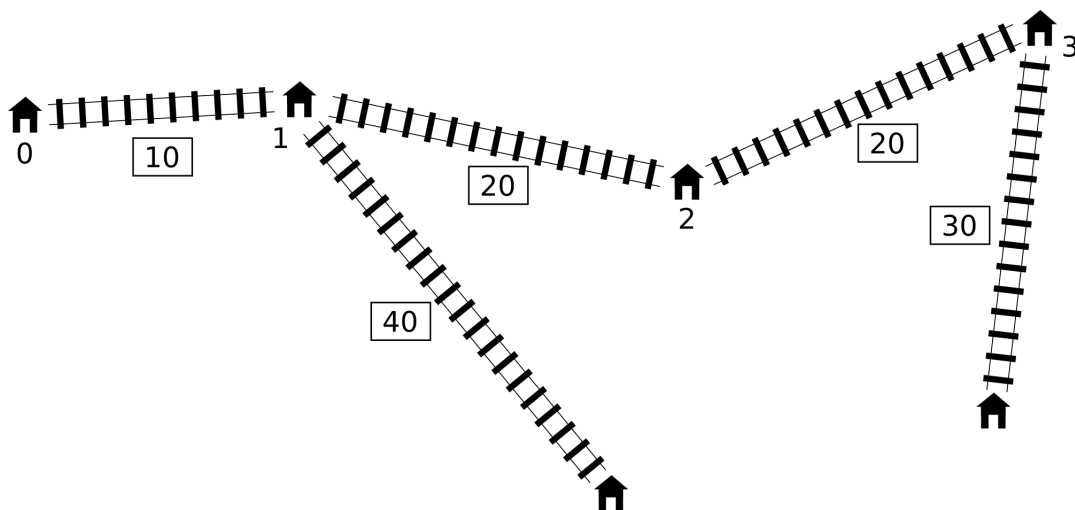


Qisqa yo'l

Pavelning o'yinchoq temir yo'li bor. U juda ham sodda. n ta bekatdan iborat yagona asosiy liniya bor. Va ular nol dan $n-1$ gacha bo'lgan tartibda raqamlangan. Nol va $n-1$ bekatlar asosiy yo'lning ikki oxirida yotadi. i va $i+1$ bekatlar orasidagi masofa li sm ga teng. ($0 < i < n-1$).

Asosiy liniyadan tashqari u yerda ikkilamchi liniyalar ham bo'lishi mumkin. Har bir ikkilamchi liniya asosiy liniyadagi bekat va asosiy liniyada joylashmagan ya'ni bekat orasidagi temir yo'l kesmasidan iborat. (Bu yangi bekatlar raqamlanmagan). Asosiy liniyadagi har bir bekatdan 1 tadan ortiq bo'lmagan ikkilamchi liniya boshlanishi mumkin. i bekatdan boshlangan ikkilamchi liniyaning uzunligi di sm ga teng. i bekatdan boshlanuvchi ikkilamchi yo'l yo'qligini bildirish uchun $di=0$ deymiz.



Endi Pavel temir yo'lning bitta qo'shimcha kesmasini qurmoqchi: bu asosiy yo'lda joylashgan ikkita turli (qo'shni ham bo'lishi mumkin) bekatlar orasidagi tezkor yo'l. Tezkor yo'l u qanday ikki bekatni tutashtirishdan qat'iy nazar aniq c sm uzunlikka ega bo'lishi kerak.

Temir yo'lning har bir kesmasi, jumladan, yangi tezkor yo'l ham ikkita yo'nalishda ishlatilishi mumkin. Ikkita bekat orasidagi masofa deb , temir yo'l kesmalari bo'ylab o'tuvchi va bu ikki bekatni tutashtiruvchi minimal yo'l uzunligiga aytiladi. Temir yo'l diametri deb bekatlar juftligi orasidagi masofalarning maksimaliga aytiladi. boshqacha qilib aytganda, bu shunday eng kichik t sonki, ixtiyoriy ikki bekat orasidagi masofa t dan katta emas.

Pavel tezkor yo'lning shunday qurmoqchiki, natijaviy diametr iloji boricha kichik bo'lsin.

Realizatsiya tafsilotlari:

Siz ushbu funktsiyani yozishingiz kerak.

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

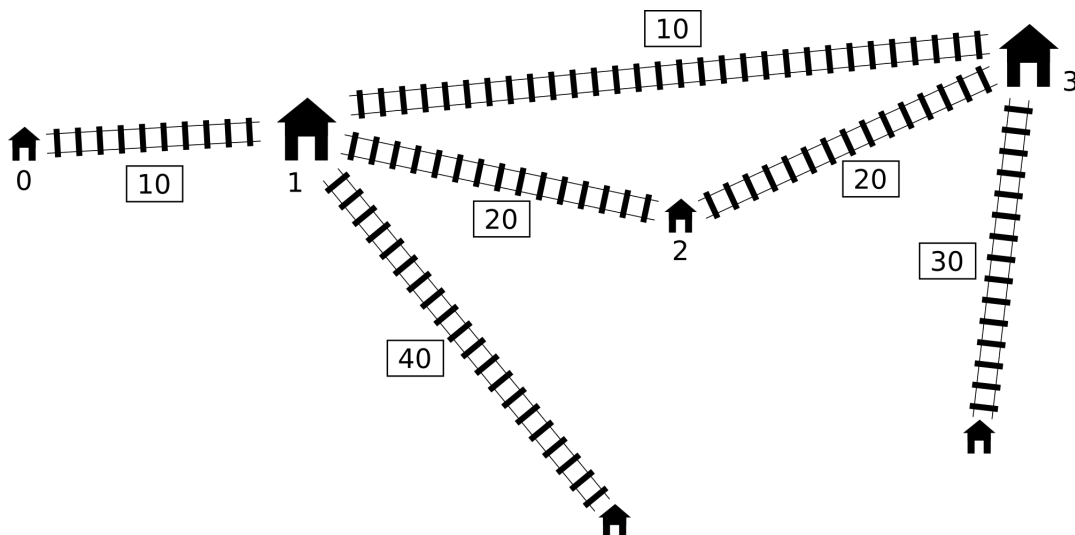
- o n: asosiy yo'ldagi bekatlar soni.
- o l: asosiy yo'ldagi bekatlar orasidagi masofa (n-1 uzunlikdagi massiv)
- o d: ikklamchi yo'llar uzunligi (n uzunlikdagi massiv)
- o c: yangi tezkor yo'l uzunligi.
- o Funktsiya tezkor yo'ni qurib bo'lgach, temir yo'lning mumkin bo'lgan minimal diametrini qaytarish kerak.
- o Ushbu shablonlardan foydalaning.

MISOLLAR

1-misol. Quyida keltirilgan rasmdagi yo'l uchun ushbu funktsiya chaqiriladi:

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

Optimal yechim – quyida keltirilganidek, 1- va 3-bekatlarda tezkor yo'l qurish.



Hosil bo'lgan temir yo'lning diametric 80 sm ga teng. Shu sababli funktsiya 80 ni qaytarishi kerak.

2-misol. Agar quyidagi funktsiyani chaqirilsa:

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

Optimal yechim 2- va 7-bekatlarni tutashtirish. Bunda, diametr 110 ga teng.

3-misol. Agar ushbu funktsiya chaqirilsa,

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

Optimal yechim 1- va 2-bekatlarni birlashtirish. Diametri 21 ga teng.

4-misol. Agar ushbu funktsiya chaqirilsa:

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

[1, 1, 1], 3)

Ixtiyoriy ikki bekatni 3 ga teng uzunlikdagi tezkor yo'l bilan tutashtirish temir yo'lning 4 ga teng bo'lgan boshlang'ich diametrini kamaytirmaydi.

Baholash tizimi:

Barcha masala ichidagi masalalarda $2 \leq n \leq 1\,000\,000$ $1 \leq l_i \leq 109$, $0 \leq d_i \leq 109$, $1 \leq c \leq 109$

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

Tekshiruvchi modul misoli: tekshiruvchi modul kiruvchi ma'lumotlarni ushbu formatda o'qiydi:

1-qator. n va c butun sonlari.

2-qator. l_0, l_1, \dots, l_{n-2} butun sonlari.

3-qator. d_0, d_1, \dots, d_{n-1} butun sonlari.