Closing Time

Vengriya 0 da $N-1$ gacha raqamlangan N ta shahardan iborat davlat.

Shaharlar 0 dan $N-2$ gacha raqamlangan jami $N-1$ ta *ikki tomonlama* yo'llar orqali bog'langan. Har bir $j(0 \leq j \leq N-2)$ uchun j - yo'l $U[j]$ va $V[j]$ - shaharlarni $W[j]$ uzunlikdagi yo'l orqali bog'laydi, ya'ni shu yo'ldan yurib o'tish uchun $W[j]$ vaqt birligi talab etiladi. Barcha yo'llar ikkita har xil shaharlarni bog'laydi va har bir juftlik orasida ko'pi bilan bitta yo'l bor.

ikkita har xil a va b shaharlarni orasidagi **path** deb har xil shaharlardan iborat $p_0, p_1, \ldots, p_t$ ketma-ketlikka aytiladiki, bunda:

- $p_0 = a$,
- $p_t = b$,
- har bir i ($0 \leq i < t$) uchun, p_i va p_{i+1} - shaharlarning orasida ikki tomonlama yo'l mavjud.

Bu yerda ixtiyoriy shahardan boshqa bir ixtiyoriy shaharga ikki tomonlama yo'llardan foydalanib sayohat qilish mumkinligi kafolotlangan, va bu yo'l har bir juftlik shahar uchun yagona ekanligi ma'lum.

$p_0, p_1, \ldots, p_t$ path ning **uzunligi** shu orada bosib o'tiladigan har bir yo'lning uzunliklari yig'indisiga teng.

Vengriyada ko'p odamlar ikki yirik shaharda tashkil etilgan Foundation Day festivalida ishtirok etish uchun sayohat qilishadi. Bayramlar tugagach ular uylariga qaytishadi. Hukumat olomonning mahalliy aholini bezovta qilishiga yo'l qo'ymaslikni xohlaydi, shuning uchun ular ma'lum vaqtlarda barcha shaharlarni blokirovka qilishni rejalashtirmoqda. Har bir shaharga hukumat tomonidan manfiy bo'lmagan **yopilish vaqti** tayinlanadi. Hukumat barcha yopilish vaqtlarining yig'indisi K dan oshmasligi kerak degan qarorga keldi. Aniqrog‘i, 0 va $N-1$ oralig‘idagi har bir i uchun, i shahriga tayinlangan yopilish vaqti $c[i]$ manfiy bo‘lmagan butun sondir. Barcha $c[i]$ yig‘indisi K dan oshmasligi kerak. a shaharni va yopilish vaqtlarini ko'rib chiqing.

Agar $b = a$ bo'lsa yoki ular orasidagi path $p_0, \ldots, p_t$ ($p_0 = a$ va $p_t = b$) quyidagi shartlarga javob bersa biz a shahardan b shaharga **borish mumkin** deb hisoblaymiz:

- p_0, p_1 orasidagi yo'l uzunligi ko'pi bilan $c[p_1]$ bo'lsa va,
- p_0, p_1, p_2 orasidagi yo'l uzunligi ko'pi bilan $c[p_2]$ bo'lsa va,
- $\ldots$
- $p_0, p_1, p_2, \ldots, p_t$ orasidagi yo'l uzunligi ko'pi bilan $c[p_t]$.

Bu yil festival bo'lib o'tadigan shaharlar sifatida X va Y - shaharlar tanlandi. Har bir yopilish vaqtini belgilash uchun **qulaylik ball** quyidagi ikkita raqamning yig'indisi sifatida aniqlanadi:

- X shahardan yetib boradigan shaharlar soni.
- Y shahardan yetib boradigan shaharlar soni.

E'tibor bering, agar bitta shaharga X shahridan va Y shahridan borish mumkin bo'lsa, u qulaylik reytingida *ikki marta* hisoblanadi.

Sizning vazifangiz yopilish vaqtini belgilash orqali erishish mumkin bo'lgan maksimal qulaylik ballini hisoblashdir.

Implement qilish uchun tafsilotlar

Siz quyidagi funksiyani implement qilishingiz kerak:

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

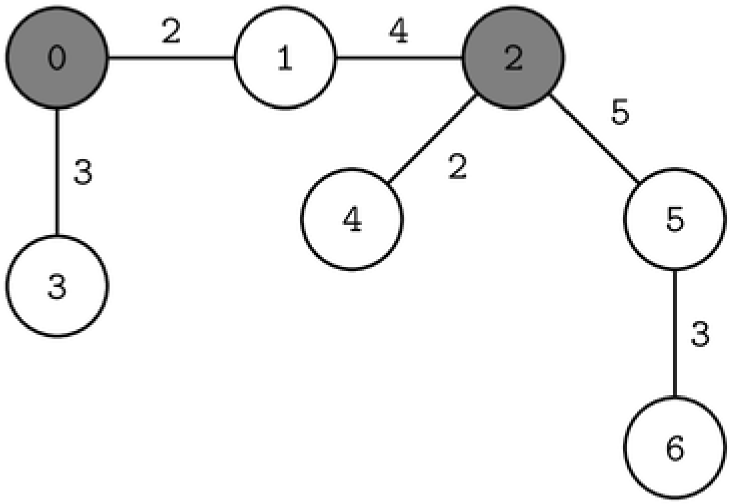
- N: shaharlar soni.
- X, Y: festival bo'lib o'tadigan shaharlar.
- K: yopilish vaqtlarining umumiy yig'indisi bo'lishi mumkin bo'lgan eng katta qiymat.
- U, V: $N-1$ uzunlikdagi shaharlarning bog'lanish yo'llarini ifodalovchi massivlar.
- W: $N-1$ uzunlikdagi har bir yo'lning uzunligini ifodalovchi massiv.
- Bu funksiya yopilish vaqtini belgilash orqali erishish mumkin bo'lgan maksimal qulaylik ballini qaytarishi kerak.
- Har bir test case da bu funksiyaga **bir necha marotaba** murojaat qilinishi mumkin..

Example

Consider the following call:

```
max_score(7, 0, 2, 10,
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

This corresponds to the following road network:



Suppose the closing times are assigned as follows:

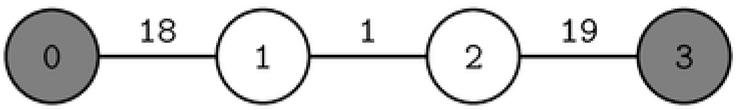
City	0	1	2	3	4	5	6
Closing time	0	4	0	3	2	0	0

Note that the sum of all closing times is 9, which is not more than $K = 10$. Cities 0, 1, and 3 are reachable from city X ($X = 0$), while cities 1, 2, and 4 are reachable from city Y ($Y = 2$). Therefore, the convenience score is $3 + 3 = 6$. There is no assignment of closing times with convenience score more than 6, so the procedure should return 6.

Also consider the following call:

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

This corresponds to the following road network:



Suppose the closing times are assigned as follows:

City	0	1	2	3
Closing time	0	1	19	0

City 0 is reachable from city X ($X = 0$), while cities 2 and 3 are reachable from city Y ($Y = 3$). Therefore, the convenience score is $1 + 2 = 3$. There is no assignment of closing times with convenience score more than 3, so the procedure should return 3.

Constraints

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (for each j such that $0 \leq j \leq N-2$)
- $1 \leq W[j] \leq 10^6$ (for each j such that $0 \leq j \leq N-2$)
- It is possible to travel from any city to any other city by using the roads.
- $S_N \leq 200\,000$, where S_N is the sum of N over all calls to `max_score` in each test case.

Subtasks

We say that a road network is **linear** if road i connects cities i and $i+1$ (for each i such that $0 \leq i \leq N-2$).

1. (8 points) The length of the path from city X to city Y is greater than $2K$.
2. (9 points) $S_N \leq 50$, the road network is linear.
3. (12 points) $S_N \leq 500$, the road network is linear.
4. (14 points) $S_N \leq 3\,000$, the road network is linear.
5. (9 points) $S_N \leq 20$
6. (11 points) $S_N \leq 100$
7. (10 points) $S_N \leq 500$
8. (10 points) $S_N \leq 3\,000$
9. (17 points) No additional constraints.

Sample Grader

Let C denote the number of scenarios, that is, the number of calls to `max_score`. The sample grader reads the input in the following format:

- line 1: C

The descriptions of C scenarios follow.

The sample grader reads the description of each scenario in the following format:

- line 1: $N \; X \; Y \; K$
- line $2+j$ ($0 \leq j \leq N-2$): $U[j] \; V[j] \; W[j]$

The sample grader prints a single line for each scenario, in the following format:

- line 1: the return value of `max_score`