



Час локдауну

Угорщина - це країна, в якій N міст, які пронумеровані від 0 до $N - 1$.

Міста з'єднані $N - 1$ двосторонніми дорогами, які пронумеровані від 0 до $N - 2$. Для кожного j , що $0 \leq j \leq N - 2$, дорога j з'єднує місто $U[j]$ та місто $V[j]$, а також має довжину $W[j]$, тобто, дорогу між містами можна пройти за $W[j]$ одиниць часу. Кожна дорога з'єднує два різні міста, а також кожна пара міст з'єднана не більш ніж однією дорогою.

Шлях між двома різними містами a та b - це послідовність p_0, p_1, \dots, p_t різних міст таких, що:

- $p_0 = a$,
- $p_t = b$,
- для кожного i ($0 \leq i < t$) є дорога між містами p_i та p_{i+1} .

Відомо, що можливо пройти між будь-якою парою різних міст, використовуючи наявні дороги. Іншими словами, існує шлях між будь-якою парою різних міст. Можна показати, що цей шлях унікальний для будь-якої пари різних міст.

Довжина шляху p_0, p_1, \dots, p_t - це сумарна довжина t доріг, які поєднують послідовні міста на шляху.

Багато людей хочуть відвідати фестивалі Дня заснування Угорщини, які проходять в деяких великих містах. Люди повертаються додому, коли святкування закінчуються. Уряд не хоче, щоб цей натовп заважав місцевим жителям, тому планує заблокувати деякі міста (організувати локдаун) в певні моменти часу. Кожному місту урядом буде присвоєно певний невід'ємний **час локдауну**. Уряд вирішив, що сумарний час всіх локдаунів, не має перевищувати K . Тобто, для кожного i між 0 та $N - 1$ включно час локдауну, присвоєний місту i , - це невід'ємне ціле число $c[i]$. Сума всіх $c[i]$ не має перевищувати K .

Розглянемо місто a і деяке присвоєння часу локдауну. Скажемо, що місто b **доступне** з міста a тоді і лише тоді, коли або $b = a$, або шлях p_0, \dots, p_t між цими двома містами (зокрема, коли $p_0 = a$ та $p_t = b$) задовольняє наступні умови:

- довжина шляху p_0, p_1 не перевищує $c[p_1]$ та
- довжина шляху p_0, p_1, p_2 не перевищує $c[p_2]$ та
- ...
- довжина шляху $p_0, p_1, p_2, \dots, p_t$ не перевищує $c[p_t]$.

Цього року два основні фестивалі знаходяться в містах X та Y . Для кожного присвоєння часу локдауну, **оцінка зручності** визначається як сума двох чисел:

- Кількість міст, які доступні з міста X .
- Кількість міст, які доступні з міста Y .

Зверніть увагу, якщо місто доступне з міста X , а також і з міста Y , то воно рахується *двічі* в оцінці зручності.

Вам потрібно знайти максимально можливу оцінку зручності, яку можна досягнути шляхом присвоєння часу локдауну.

Деталі імплементації

Вам потрібно імплементувати наступну функцію:

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

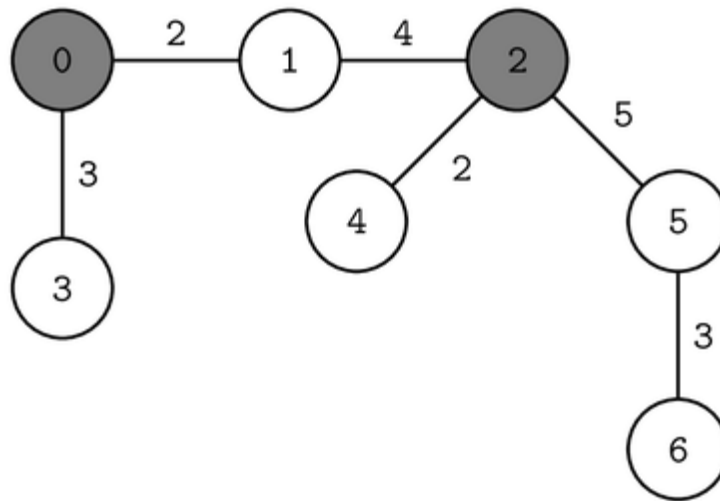
- N : кількість міст.
- X, Y : міста, в яких проходять основні фестивалі.
- K : верхнє обмеження на сумарний час локдауну.
- U, V : масиви довжини $N - 1$, які позначають міста, з'єднані дорогою.
- W : масив довжини $N - 1$, який позначає довжини доріг.
- Ця функція має повернути максимально можливу оцінку зручності, яку можна досягнути шляхом присвоєння часу локдауну.
- Ця функція може викликатися **кілька разів** в одному тесті.

Приклад

Розглянемо наступний виклик:

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

Він відповідає такій мережі доріг:



Припустимо, що час локдауну присвоєно таким чином:

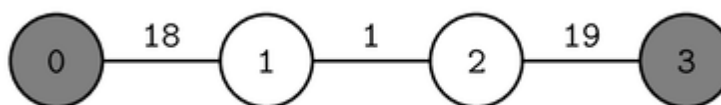
Місто	0	1	2	3	4	5	6
Час локдауну	0	4	0	3	2	0	0

Зверніть увагу, що сумарний час локдауну дорівнює 9, тобто, не перевищує $K = 10$. Міста 0, 1 та 3 доступні з міста X ($X = 0$), а міста 1, 2 та 4 доступні з міста Y ($Y = 2$). Тому, оцінка зручності рівна $3 + 3 = 6$. Немає присвоєння часу локдауну, де оцінка зручності перевищувала б 6, тому функція має повернути 6.

Також розглянемо наступний виклик:

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

Він відповідає такій мережі доріг:



Припустимо, що час закриття присвоєно таким чином:

Місто	0	1	2	3
Час локдауну	0	1	19	0

Місто 0 доступне з міста X ($X = 0$), а міста 2 та 3 доступні з міста Y ($Y = 3$). Тому, оцінка зручності $1 + 2 = 3$. Немає присвоєння часу локдауну, де оцінка зручності перевищувала б 3, тому функція має повернути 3.

Обмеження

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (для кожного j такого, що $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (для кожного j такого, що $0 \leq j \leq N - 2$)
- Можливо пройти з будь-якого міста в будь-яке інше, використовуючи дороги.
- $S_N \leq 200\,000$, де S_N - це сума N в усіх викликах `max_score`.

Підзадачі

Ми назвемо мережу доріг **лінійною**, якщо дорога i з'єднує міста i та $i + 1$ (для всіх i таких, що $0 \leq i \leq N - 2$).

1. (8 балів) Довжина шляху з міста X до міста Y перевищує $2K$.
2. (9 балів) $S_N \leq 50$, мережа доріг лінійна.
3. (12 балів) $S_N \leq 500$, мережа доріг лінійна.
4. (14 балів) $S_N \leq 3\,000$, мережа доріг лінійна.
5. (9 балів) $S_N \leq 20$
6. (11 балів) $S_N \leq 100$
7. (10 балів) $S_N \leq 500$
8. (10 балів) $S_N \leq 3\,000$
9. (17 балів) Без додаткових обмежень.

Приклад градера

Нехай C позначає кількість сценаріїв, тобто, кількість викликів функції `max_score`. Градер зчитує дані у такому форматі:

- рядок 1: C

Нижче наведено опис C сценаріїв.

Градер зчитує дані кожного сценарію у наступному форматі:

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

Градер виводить дані кожного сценарію у наступному форматі:

- рядок 1: число, яке повернула функція `max_score`