Време на затваряне

Унгария е държава с N града, номерирани с числата от 0 до $N-1$.

Градовете са свързани с $N-1$ *двупосочни* шосета, номерирани с числата от 0 до $N-2$. За всяко j, такова че $0 \le j \le N-2$, шосе с номер j свързва градовете с номера $U[j]$ и $V[j]$ и е с дължина $W[j]$, т.е. позволява да се пътува между градовете за време $W[j]$. Всяко шосе свързва два различни града и всяка двойка градове е свързана с най-много едно шосе.

Път между два различни града a и b е редица от различни градове $p_0, p_1, \ldots, p_t$, така че:

- $p_0 = a$,
- $p_t = b$,
- за всяко i ($0 \le i < t$) има шосе между градовете p_i и p_{i+1}.

Възможно е да се пътува от всеки град до всеки друг град, като се използват шосетата, т.е. има път между всеки два града. Може да се покаже, че този път е уникален за всяка двойка различни градове.

Дължината на пътя $p_0, p_1, \ldots, p_t$ е равна на сумата от дължините на t-те шосета, свързващи последователните градове по пътя.

В Унгария много хора пътуват, за да посетят тържествата за Деня на основаването в два големи града. Веднъж, когато приключат празненствата, гражданите се връщат обратно по домовете си. Правителството иска групата от хора да не притеснява местните, затова те планират да затворят градовете в определени времена. На всеки град ще бъде зададено неотрицателно цяло число от правителството, което ще е **време на затваряне** на града. Освен това е решено, че сумата от времената на затваряне не трябва да е повече от K. По-точно за всяко i от 0 до $N-1$, включително, времето на затваряне на град i е неотрицателно цяло число $c[i]$ и сумата на всички $c[i]$ е по-малка или равна на K.

Нека разгледаме град a и някакво разпределение на времената на затваряне на градовете. Казваме, че град b е **достижим** от град a тогава и само тогава, когато $b = a$ или пътят $p_0, \ldots, p_t$ между тези два града (разбира се $p_0 = a$ и $p_t = b$) спазва следните условия:

- дължината на пътя p_0, p_1 е най-много $c[p_1]$, и
- дължината на пътя p_0, p_1, p_2 е най-много $c[p_2]$, и
- $\ldots$
- дължината на пътя $p_0, p_1, p_2, \ldots, p_t$ е най-много $c[p_t]$.

Тази година двата града с тържества са град с номер X и град с номер Y. За дадено разпределение на времената на затваряне **оценката за удобство** е дефинирана като сумата от следните две числа

- Броят градове, които са достижими от град X.
- Броят градове, които са достижими от град Y.

Обърнете внимание, че ако град е достижим и от град X, и от град Y, то той се брои *два* пъти в оценката за удобство.

Вашата задача е да намерите максималната възможна оценка за удобство, която може да бъде постигната от някакво разпределение на времената на затваряне.

Детайли по имплементацията

Трябва да напишете следната функция.

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

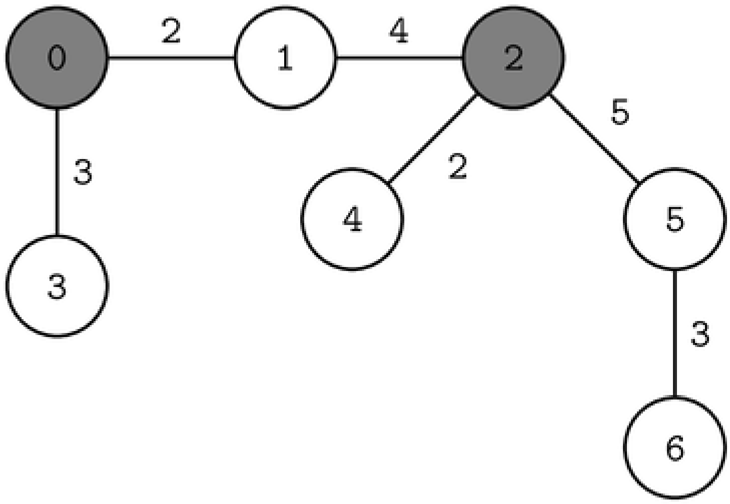
- N: броят на градовете.
- X, Y: градовете с тържества.
- K: горната граница на сумата от времената на затваряне.
- U, V: масиви с големини $N-1$, описващи шосетата.
- W: масив с големина $N-1$, описващ дължините на шосетата.
- Тази функция трябва да върне максималната възможна оценка за удобство, която може да бъде постигната от някое разпределение на времената на затваряне.
- Тази функция може да бъде извикана **няколко пъти** (от програмата на журито) в рамките на един и същ тест.

Пример

Нека имаме следното извикване:

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

Това съответства на следната мрежа от шосета:



Нека предположим, че времената на затваряне са зададени по следния начин:

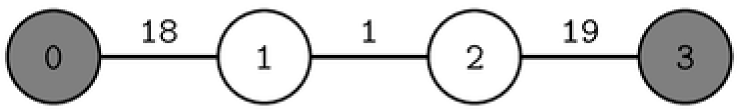
Град	0	1	2	3	4	5	6
Време на затваряне	0	4	0	3	2	0	0

Обърнете внимание, че сумата от времената на затваряне е 9, което не е повече от $K = 10$. Градовете с номера 0, 1 и 3 са достижими от град X ($X = 0$), а градовете с номера 1, 2 и 4 са достижими от град Y ($Y = 2$). Затова оценката за удобство е $3 + 3 = 6$. Няма разпределение на времената на затваряне с оценка за удобство повече от 6, затова функцията трябва да върне 6.

Нека разгледаме и следното извикване:

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

Това съответства на следната мрежа от шосета:



Нека предположим, че времената на затваряне са зададени по следния начин:

Град	0	1	2	3
Време на затваряне	0	1	19	0

Град с номер 0 е достижим от град X ($X = 0$), а градовете с номера 2 и 3 са достижими от град Y ($Y = 3$). Затова оценката за удобство е $1 + 2 = 3$. Няма разпределение на времената на затваряне с оценка за удобство повече от 3, затова функцията трябва да върне 3.

Ограничения

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (за всяко j, такова че $0 \leq j \leq N-2$)
- $1 \leq W[j] \leq 10^6$ (за всяко j, такова че $0 \leq j \leq N-2$)
- Възможно е да се пътува от всеки град до всеки друг град, като се използват шосетата.
- $S_N \leq 200\,000$, където S_N е сумата на N за всички извиквания на `max_score` в рамките на един тест.

Подзадачи

Нека да уточним, че наричаме една пътна мрежа **пръчка**, ако шосе с номер i свързва градовете с номера i и $i+1$ (за всяко i, такова че $0 \leq i \leq N-2$).

1. (8 точки) Дължината на пътя между градовете X и Y е по-голяма от $2K$.
2. (9 точки) $S_N \leq 50$, пътната мрежа е пръчка.
3. (12 точки) $S_N \leq 500$, пътната мрежа е пръчка.
4. (14 точки) $S_N \leq 3\,000$, пътната мрежа е пръчка.
5. (9 точки) $S_N \leq 20$
6. (11 точки) $S_N \leq 100$
7. (10 точки) $S_N \leq 500$
8. (10 точки) $S_N \leq 3\,000$
9. (17 точки) Няма допълнителни ограничения.

Локално тестване

Нека означим с C броя на случаите или, по друг начин казано, броя на извикванията на `max_score` в рамките на един тест. Локалният грейдър ще чете входа в следния формат:

- ред 1: C

След което се въвеждат описанията на C случая. Локалният грейдър чете описанието на всеки случай в следния формат:

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

Локалният грейдър отпечатва по един ред за всеки случай в следния формат:

- ред 1: върнатата стойност на `max_score`