Closing Time

Hongarije is een land met N steden, genummerd van 0 tot en met $N-1$.

De steden zijn verbonden door $N-1$ *tweerichtings* wegen, genummerd van 0 tot en met $N-2$. Voor elke j zodat $0 \leq j \leq N-2$, verbindt weg j de steden $U[j]$ en $V[j]$ en de weg heeft een lengte $W[j]$, wat betekent dat je $W[j]$ tijdseenheden nodig hebt om van de ene stad naar de andere te reizen. Elke weg verbindt twee verschillende steden met elkaar en ieder paar steden is maximaal door één weg verbonden.

Een **pad** tussen twee verschillende steden a en b is een rij $p_0, p_1, \ldots, p_t$ verschillende steden, zodat:

- $p_0 = a$,
- $p_t = b$,
- voor elke i ($0 \leq i < t$), is er een weg die de steden p_i en p_{i+1} met elkaar verbindt.

Het is mogelijk om van een stad naar een willekeurige andere stad te reizen door gebruik te maken van de wegen, dat wil zeggen dat er een pad bestaat tussen ieder paar steden. Je kunt bewijzen dat dit pad uniek is, voor ieder paar steden.

De **lengte** van een pad $p_0, p_1, \ldots, p_t$ is de som van de lengtes van de t wegen die de opvolgende steden langs het pad met elkaar verbinden.

In Hongarije reizen veel mensen om deel te nemen aan de feesten voor Onafhankelijkheidsdag, die worden gevierd in twee belangrijke steden. Als de feestelijkheden voorbij zijn, keren ze terug naar hun huizen. De overheid wil voorkomen dat de menigte de plaatselijke bevolking verstoort, dus is men van plan om alle steden op een bepaald moment te sluiten. Elke stad krijgt door de overheid een niet-negatieve **sluitingstijd** toebedeeld. De overheid heeft bepaald dat de som van alle sluitingstijden niet groter mag zijn dan K. Preciezer gezegd: voor elke i van 0 tot en met $N-1$, is de sluitingstijd van stad i een niet-negatieve integer $c[i]$. De som van alle $c[i]$ mag niet groter zijn dan K.

Bekijk een stad a en enkele mogelijke sluitingstijden. We zeggen dat een stad b alleen **bereikbaar** is vanuit stad a dan en slechts dan als $b = a$, of het pad $p_0, \ldots, p_t$ tussen deze twee steden (met $p_0 = a$ en $p_t = b$) voldoet aan de volgende voorwaarden:

- de lengte van het pad p_0, p_1 is hoogstens $c[p_1]$, en
- de lengte van het pad p_0, p_1, p_2 is hoogstens $c[p_2]$, en

- ...
- de lengte van het pad $p_0, p_1, p_2, \ldots, p_t$ is hoogstens $c[p_t]$.

Dit jaar zijn de belangrijkste feesten in de steden X en Y. Voor elke toewijzing van sluitingstijden, is de **gemaksscore** gedefinieerd als de som van de volgende twee getallen:

- Het aantal steden dat kan worden bereikt vanuit stad X.
- Het aantal steden dat kan worden bereikt vanuit stad Y.

Merk op dat als een stad bereikbaar is vanuit stad X en vanuit stad Y, dat hij dan tweemaal meetelt in de gemaksscore.

Jouw programma moet de maximale gemaksscore vaststellen die kan worden bereikt voor de één of andere toewijzing van sluitingstijden.

Implementatiedetails

Jij moet de volgende functie implementeren.

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

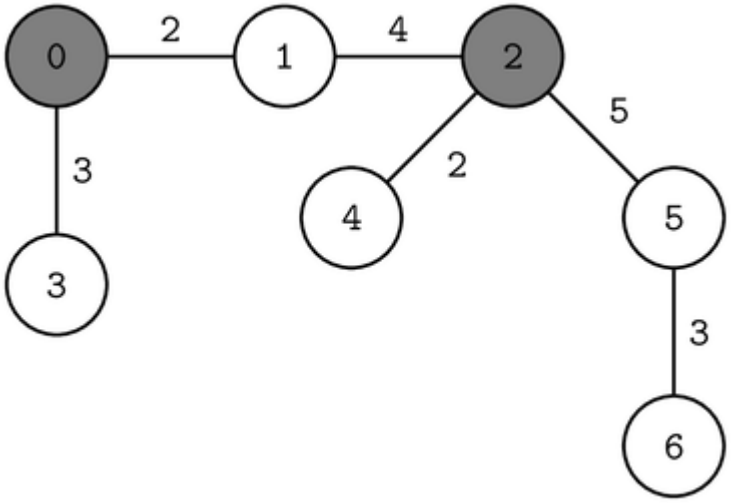
- N: het aantal steden.
- X, Y: de steden met de belangrijkste festiviteiten.
- K: het maximum van de som van de sluitingstijden.
- U, V: arrays van de plaatsen die door de $N-1$ wegen worden verbonden.
- W: array van lengtes van de $N-1$ wegen.
- Deze functie retourneert de maximale gemaksscore die kan worden bereikt door de sluitingstijden geschikt te kiezen.
- Deze functie mag **zo vaak als nodig** worden aangeroepen in elke testcase.

Voorbeeld

Bekijk de volgende aanroep:

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

Dit hoort bij het volgende wegennet:



Stel dat de sluitingstijden als volgt zijn toegewezen:

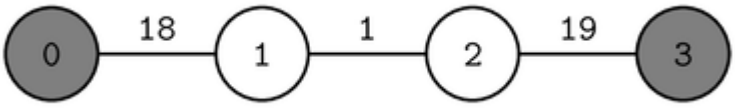
Stad	0	1	2	3	4	5	6
Sluitingstijd	0	4	0	3	2	0	0

Merk op dat de som van alle sluitingstijden 9 is, dat is niet meer dan $K = 10$. De steden 0, 1, en 3 zijn bereikbaar vanuit stad X ($X = 0$), terwijl de steden 1, 2, en 4 bereikbaar zijn vanuit stad Y ($Y = 2$). De gemaksscore is daarom $3 + 3 = 6$. Er is geen manier waarbij de sluitingstijden zo kunnen worden gekozen dat de gemaksscore hoger wordt dan 6, dus de functie moet 6 retourneren.

Bekijk ook de volgende aanroep:

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

Dit komt overeen met het volgende wegennet:



Stel dat de sluitingstijden als volgt worden toegewezen:

Stad	0	1	2	3
Sluitingstijd	0	1	19	0

Stad 0 is bereikbaar vanuit stad X ($X = 0$), terwijl de steden 2 en 3 bereikbaar zijn vanuit stad Y ($Y = 3$). Daarom is de gemaksscore $1 + 2 = 3$. Er is geen manier waarbij de sluitingstijden zo kunnen worden gekozen dat de gemaksscore hoger wordt dan 3, dus de functie moet 3 retourneren.

Randvoorwaarden

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (voor elke j met $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (voor elke j met $0 \leq j \leq N - 2$)
- Alle steden zijn onderling bereikbaar via de wegen.
- $S_N \leq 200\,000$, waarbij S_N de som is van N over alle aanroepen van `max_score` in iedere testcase.

Subtasks

Een wegennet heet **lineair** als weg i de steden i en $i + 1$ verbindt (voor elke i met $0 \leq i \leq N - 2$).

1. (8 punten) De lengte van het pad van stad X naar stad Y is groter dan $2K$.
2. (9 punten) $S_N \leq 50$, het wegennet is lineair.
3. (12 punten) $S_N \leq 500$, het wegennet is lineair.
4. (14 punten) $S_N \leq 3\,000$, het wegennet is lineair.
5. (9 punten) $S_N \leq 20$
6. (11 punten) $S_N \leq 100$
7. (10 punten) $S_N \leq 500$
8. (10 punten) $S_N \leq 3\,000$
9. (17 punten) Geen aanvullende randvoorwaarden.

Sample Grader

Stel dat er C verschillende scenario's zijn, dat is het maximale aantal aanroepen van `max_score`. De sample grader leest de invoer in het volgende format:

- regel 1: C

Dan volgen beschrijvingen van C scenario's.

De sample grader leest de beschrijving van elk scenario in het volgende format:

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

De sample grader print een enkele regel voor ieder scenario, in het volgende format:

- regel 1: de geretourneerde waarde van `max_score`