Sperrzeit

Ungarn ist ein Land mit N Städten. Diese Städte sind von 0 bis $N-1$ nummeriert.

Die Städte sind durch $N-1$ Straßen verbunden, die in *beide Richtungen befahrbar sind* und von 0 bis $N-2$ nummeriert sind. Für jedes j mit $0 \leq j \leq N-2$ verbindet Straße j Stadt $U[j]$ mit Stadt $V[j]$ und hat die Länge $W[j]$. Das heißt, es werden $W[j]$ Zeiteinheiten benötigt, um von einer Stadt zur anderen zu fahren. Jede Straße verbindet zwei verschiedene Städte und jedes Paar von Städten ist mit maximal einer Straße verbunden.

Ein **Pfad** zwischen zwei verschiedenen Städten a und b ist eine Sequenz $p_0, p_1, \ldots, p_t$ unterschiedlicher Städte, so dass:

- $p_0 = a$,
- $p_t = b$,
- Für jedes i mit $0 \leq i < t$ existiert eine Straße welche die Städte p_i und p_{i+1} verbindet.

Es ist möglich, von jeder Stadt aus in jede andere über das Straßennetz zu reisen; das heißt, es gibt immer einen Pfad zwischen je zwei verschiedenen Städten. Man kann zeigen, dass dieser Pfad für jedes Paar verschiedener Städte eindeutig ist.

Die **Länge** eines Pfades $p_0, p_1, \ldots, p_t$ ist die Summe der Längen der t Straßen, welche die aufeinanderfolgenden Städte verbinden.

In Ungarn fahren viele Leute zum Staatsfeiertag zu den Hauptfestplätzen in zwei Großstädte. Sobald die Feierlichkeiten vorüber sind, fahren sie nach Hause. Die Regierung will verhindern, dass die Menschenmasse die lokalen Einwohner stört. Deshalb plant sie, alle Städte an bestimmten Zeiten zu sperren. Jeder Stadt wird eine nicht negative **Sperrzeit** von der Regierung zugewiesen. Die Regierung hat entschieden, dass die Summe aller Sperrszeiten nicht größer als K sein darf. Genauer gesagt, für alle i zwischen 0 und $N-1$ inklusive, ist die Sperrzeit von Stadt i eine nicht negative Ganzzahl $c[i]$. Die Summe aller $c[i]$ darf nicht größer als K sein.

Nehmen wir eine Stadt a und eine gegebene Zuweisung von Sperrzeiten. Stadt b ist von Stadt a aus **erreichbar** genau dann wenn $b = a$ oder wenn der Pfad $p_0, \ldots, p_t$ zwischen beiden Städten (also insbesondere $p_0 = a$ und $p_t = b$) folgende Bedingungen erfüllt:

- Die Länge des Pfades p_0, p_1 ist nicht größer als $c[p_1]$, und
- Die Länge des Pfades p_0, p_1, p_2 ist nicht größer als $c[p_2]$, und
- ...

- Die Länge des Pfades $p_0, p_1, p_2, \ldots, p_t$ ist nicht größer als $c[p_t]$.

Dieses Jahr sind die Hauptfestplätze in Stadt X und Stadt Y. Für jede Zuweisung von Sperrzeiten ist die **Komfortpunktzahl** als die Summe zweier Zahlen definiert:

- Die Anzahl an Städten, die von Stadt X aus erreichbar sind.
- Die Anzahl an Städten, die von Stadt Y aus erreichbar sind.

Beachte, dass eine Stadt, die von Stadt X und von Stadt Y aus erreichbar ist, für die Komfortpunktzahl *doppelt* zählt.

Deine Aufgabe ist es, die maximale Komfortpunktzahl über alle Zuweisungen von Sperrzeiten zu berechnen.

Implementierungsdetails

Implementiere die folgende Funktion:

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

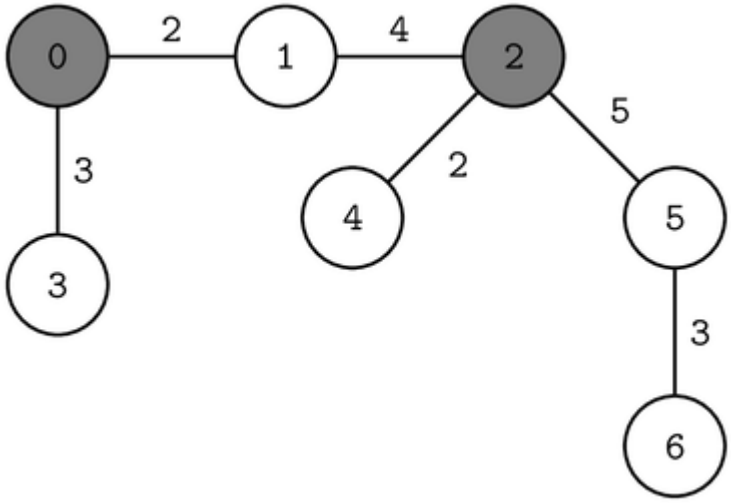
- N: die Anzahl an Städten.
- X, Y: die Städte der Hauptfestplätze.
- K: die obere Schranke der Summe der Sperrzeiten.
- U, V: Arrays der Länge $N-1$, die das Straßennetz beschreiben.
- W: Array der Länge $N-1$, das die Straßenlängen enthält.
- Diese Funktion soll die maximale Komfortpunktzahl zurückgeben, welche durch Wahl gültiger Sperrzeiten erreicht werden kann.
- Diese Funktion kann in jedem Testfall **mehrfach** aufgerufen werden.

Beispiel

Betrachte den folgenden Funktionsaufruf:

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

Dieser entspricht dem folgenden Straßennetz:



Nehmen wir an, dass die Sperrzeiten wie folgt sind:

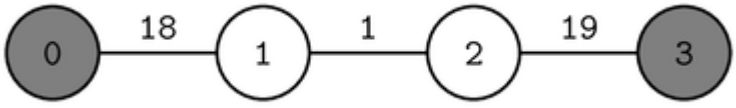
Stadt	0	1	2	3	4	5	6
Sperrzeit	0	4	0	3	2	0	0

Wir bemerken, dass die Summe aller Sperrzeiten 9 ist, was nicht mehr als $K = 10$ ist. Die Städte 0, 1 und 3 sind von Stadt X aus erreichbar (wobei $X = 0$), und die Städte 1, 2 und 4 sind von Stadt Y aus erreichbar (wobei $Y = 2$). Folglich ist die Komfortpunktzahl $3 + 3 = 6$. Es gibt außerdem keine gültigen Sperrzeiten mit Komfortpunktzahl größer als 6, somit sollte die Funktion 6 zurückgeben.

Betrachte nun den folgenden Funktionsaufruf:

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

Dieser entspricht dem folgenden Straßennetz:



Nehmen wir an, dass die Sperrzeiten wie folgt sind:

Stadt	0	1	2	3
Sperrzeit	0	1	19	0

Die Stadt 0 ist als einzige Stadt von Stadt X aus erreichbar ($X = 0$), und Städte 2 und 3 sind von Stadt Y aus erreichbar ($Y = 3$). Folglich ist die Komfortpunktzahl $1 + 2 = 3$. Es gibt außerdem

keine gültigen Sperrzeiten mit Komfortpunktzahl größer als 3, somit sollte die Funktion 3 zurückgeben.

Einschränkungen

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (für jedes j mit $0 \leq j \leq N-2$)
- $1 \leq W[j] \leq 10^6$ (für jedes j mit $0 \leq j \leq N-2$)
- Es ist möglich, von jeder Stadt zu jeder anderen Stadt über das Straßennetz zu reisen.
- $S_N \leq 200\,000$, wobei S_N die Summen aller N über alle Aufrufe von `max_score` ist, für einen gegebenen Testfall.

Teilaufgaben

Wir nennen ein Straßennetz **linear** genau dann wenn Straße i die Städte i und $i+1$ miteinander verbindet (für jedes i sodass $0 \leq i \leq N-2$).

1. (8 Punkte) Die Länge des Pfades von Stadt X nach Y ist größer als $2K$.
2. (9 Punkte) $S_N \leq 50$, das Straßennetz ist linear.
3. (12 Punkte) $S_N \leq 500$, das Straßennetz ist linear.
4. (14 Punkte) $S_N \leq 3\,000$, das Straßennetz ist linear.
5. (9 Punkte) $S_N \leq 20$
6. (11 Punkte) $S_N \leq 100$
7. (10 Punkte) $S_N \leq 500$
8. (10 Punkte) $S_N \leq 3\,000$
9. (17 Punkte) Keine weiteren Einschränkungen.

Beispielgrader

Sei C die Anzahl an Szenarien, das heißt, die Anzahl der Aufrufe von `max_score`. Der Beispielgrader liest die Eingabe im folgenden Format:

- Zeile 1: C

Die Beschreibung von C Szenarien folgt.

Der Beispielgrader liest die Beschreibung der einzelnen Szenarien im folgenden Format:

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

Der Beispielgrader gibt pro Szenario eine einzige Zeile aus im folgenden Format:

- Zeile 1: der Rückgabewert von `max_score`