Closing Time

Hungría es un país con N ciudades, enumeradas desde 0 a $N-1$.

Las ciudades están conectadas por $N-1$ calles *bidireccionales*, enumeradas desde 0 hasta $N-2$. Para cada j tal que $0 \le j \le N-2$, la calle j conecta la ciudad $U[j]$ y a la ciudad $V[j]$ y su distancia es $W[j]$, esto es, permite un viaje entre dichas ciudades en $W[j]$ unidades de tiempo. Cada calle conecta a dos ciudades diferentes, y cada par de ciudades están conectadas por a lo sumo una calle.

Un **camino** entre dos ciudades diferentes a y b es una secuencia $p_0, p_1, \ldots, p_t$ de distintas ciudades, tal que:

- $p_0 = a$,
- $p_t = b$,
- para cada i ($0 \le i < t$), hay una calle conectando a las ciudades p_i and p_{i+1}.

Es posible viajar desde una ciudad hacia cualquier otra ciudad usando las calles, esto es, existe un camino entre cada dos ciudades diferentes. Nota que este camino es único para cada par de ciudades distintas.

La **longitud** de un camino $p_0, p_1, \ldots, p_t$ es la suma de las longitudes de las t calles que conectan ciudades consecutivas a lo largo del camino.

En Hungría, mucha gente viaja para asistir a las festividades del Día de la Fundación en alguna de las ciudades mas grandes. Una vez que las celebraciones han concluido, la gente debe retornar a sus hogares. El gobierno quiere evitar que la multitud moleste a los lugareños, por lo que planea cerrar las ciudades en determinados momentos. El gobierno asignará a cada ciudad una **hora de cierre** no negativa. El gobierno ha decidido que la suma de todos los tiempos de cierre no debe ser mayor a K. Más precisamente, para cada i entre 0 and $N-1$, inclusive, el tiempo de cierre asignado para la ciudad i es un entero no negativo $c[i]$. La suma de todos los $c[i]$ no debe ser mayor que K.

Considera una ciudad a y algunas asignaciones de hora de cierre. Nosotros decimos que una ciudad b es **alcanzable** desde la ciudad a si y solo si $b = a$, o el camino $p_0, \ldots, p_t$ entre esas dos ciudades (en particular $p_0 = a$ y $p_t = b$) satisface las siguientes condiciones:

- la longitud del camino p_0, p_1 es a lo sumo $c[p_1]$, y
- la longitud del camino p_0, p_1, p_2 es a lo sumo $c[p_2]$, y

- $\ldots$
- la longitud del camino $p_0, p_1, p_2, \ldots, p_t$ es a lo sumo $c[p_t]$.

Este año, los dos mayores festivales están ubicados en la ciudad X y la ciudad Y. Por cada asignación de tiempo de cierre, el **puntaje conveniente** se define como la suma de los siguientes dos números:

- El número de ciudades alcanzables desde la ciudad X.
- El número de ciudades alcanzables desde la ciudad Y.

Observa que si una ciudad es alcanzable desde la ciudad X y alcanzable desde la ciudad Y, cuenta *dos veces* para el puntaje conveniente.

Tu tarea es calcular el máximo puntaje conveniente que se puede obtener mediante alguna asignación de horarios de cierre.

Detalles de Implementación

Tu debes implementar el siguiente procedimiento.

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

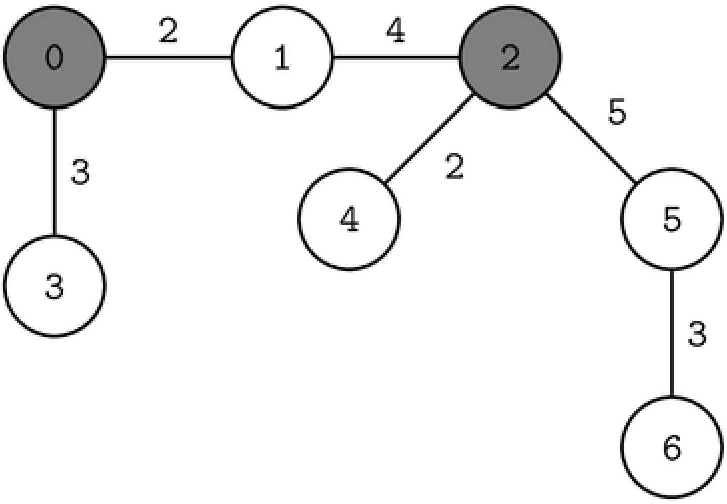
- N: el número de ciudades.
- X, Y: las ciudades con los mayores festivales.
- K: el límite superior de la suma de las horas de cierre.
- U, V: arreglos de longitud $N-1$ describiendo las calles conectadas.
- W: arreglo de longitud $N-1$ describiendo la longitud de las calles.
- Este procedimiento debería retornar el máximo puntaje conveniente que puede ser obtenido por alguna asignación de tiempo.
- Este procedimiento puede ser llamado **múltiples veces** en cada caso de prueba.

Ejemplo

Considera la siguiente llamada:

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

Esto corresponde con la siguiente red de calles:



Supóngase la siguiente asignación de tiempos de cierre:

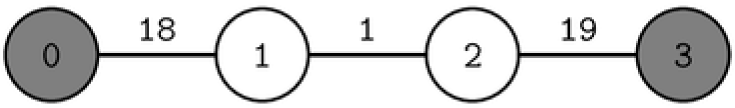
Ciudad	0	1	2	3	4	5	6
Tiempos de cierre	0	4	0	3	2	0	0

Observa que la suma de todos los tiempos de cierre es 9, que no es más que $K = 10$. Las ciudades 0, 1, y 3 son alcanzables desde la ciudad X ($X = 0$), mientras que las ciudades 1, 2, y 4 son alcanzables desde la ciudad Y ($Y = 2$). Por lo tanto, el puntaje conveniente es $3 + 3 = 6$. No hay asignación de horarios de cierre con una puntuje conveniente superior a 6, por lo que el procedimiento debería devolver 6.

También considere la siguiente llamada:

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

Esto corresponde a la siguiente red de calles:



Supóngase la siguiente asignación de tiempos de cierre:

Ciudad	0	1	2	3
Tiempos de cierre	0	1	19	0

La ciudad 0 es alcanzable desde la ciudad X ($X = 0$), mientras que las ciudades 2 y 3 son alcanzables desde la ciudad Y ($Y = 3$). Por lo tanto, el puntaje conveniente es $1 + 2 = 3$. No hay asignación de horarios de cierre con un puntaje conveniente superior a 3, por lo que el procedimiento debería devolver 3.

Restricciones

- $2 \le N \le 200\,000$
- $0 \le X < Y < N$
- $0 \le K \le 10^{18}$
- $0 \le U[j] < V[j] < N$ (por cada j tal que $0 \le j \le N - 2$)
- $1 \le W[j] \le 10^6$ (por cada j tal que $0 \le j \le N - 2$)
- Es posible viajar de cualquier ciudad a cualquier otra ciudad utilizando las calles.
- $S_N \le 200\,000$, donde S_N es la suma de N sobre todas las llamadas a `max_score`.

Subtareas

Decimos que una red de calles es **lineal** si la calle i conecta las ciudades i e $i + 1$ (por cada i tal que $0 \le i \le N - 2$)

1. (8 puntos) La longitud del camino desde la ciudad X hacia Y es mayor que $2K$.
2. (9 puntos) $S_N \le 50$, la red de calles es lineal.
3. (12 puntos) $S_N \le 500$, la red de calles es lineal.
4. (14 puntos) $S_N \le 3\,000$, la red de calles es lineal.
5. (9 puntos) $S_N \le 20$.
6. (11 puntos) $S_N \le 100$.
7. (10 puntos) $S_N \le 500$.
8. (10 puntos) $S_N \le 3\,000$.
9. (17 puntos) Sin restricciones.

Evaluador de Ejemplo

Sea C el número de escenarios, es decir, el número de llamadas a `max_score`. El evaluador lee la entrada en el siguiente formato:

- línea 1: C

Las descripciones de C escenarios siguen.

El evaluador lee la descripcion de cada escenario en el siguiente formato:

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

El evaluador imprime la descripción de cada escenario en el siguiente formato:

- línea 1: el valor de retorno de `max_score`