Hora de cierre

Hungría es un país con N ciudades, numeradas de 0 a $N-1$.

Las ciudades están conectadas por $N-1$ carreteras *bidireccionales*, numeradas de 0 a $N-2$. Para cada j tal que $0 \leq j \leq N-2$, la carretera j conecta a las ciudades $U[j]$ y $V[j]$ y tiene de longitud $W[j]$, esto es, permite a uno viajar entre las ciudades en $W[j]$ unidades de tiempo. Cada carretera conecta dos ciudades distintas, y cada par de ciudades está conectado por un máximo de una carretera.

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

- $p_0 = a$,
- $p_t = b$,
- para cada i ($0 \leq i < t$), existe una carretera que conecta a las ciudades p_i y p_{i+1}.

Es posible viajar desde cualquier ciudad a cualquier otra ciudad utilizando carreteras, o sea, existe un camino entre cualesquiera dos ciudades distintas. Puede demostrarse 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 carreteras que conectan ciudades consecutivas a lo largo del camino.

En Hungría, muchas personas viajan para asistir a las festividades del Día de la Fundación en dos ciudades importantes. Al término de las celebraciones, ellas retornan a sus hogares. El gobierno quiere prevenir que la multitud moleste a los locales, así que planean cerrar todas las ciudades a ciertas horas. A cada ciudad, el gobierno le asignará una **hora de cierre** no negativa. El gobierno ha decidido que la suma de todas las horas de cierre no debe exceder K. De forma más precisa, para cada i entre 0 y $N-1$, inclusive, la hora de cierre que se le asigna a la ciudad i es un entero no negativo $c[i]$. La suma de todos los $c[i]$ no debe exceder K.

Considera una ciudad a y una asignación de horas de cierre. 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 estas 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, las dos festividades principales estarán ubicadas en las ciudades X y Y. Para cada asignación de hora de cierre, la **puntuación de conveniencia** 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.

Fíjate que si una ciudad es alcanzable desde la ciudad X, y también desde la ciudad Y, esta se cuenta *dos* veces para fines de la puntuación de conveniencia.

Tu tarea es computar la máxima puntuación de conveniencia que se puede conseguir a través de alguna asignación de horas de cierre.

Detalles de Implementación

Debes implementar la siguiente función.

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

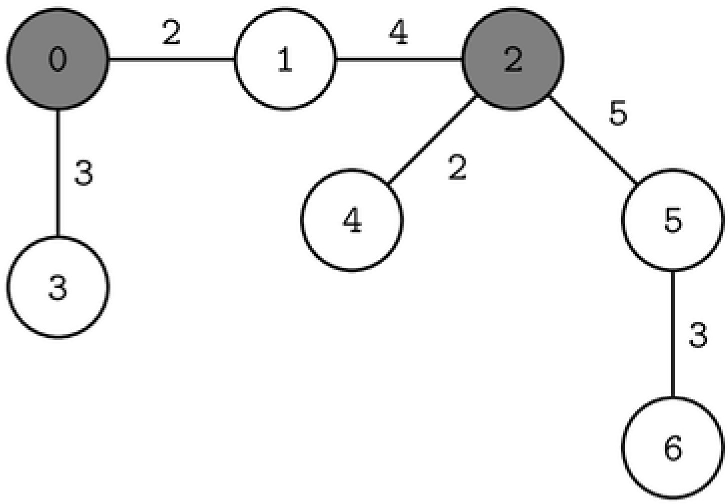
- N: el número de ciudades.
- X, Y: las ciudades donde se celebrarán las festividades principales.
- K: el límite superior de la suma de horas de cierre.
- U, V: arreglos de longitud $N-1$ que describen las conexiones sobre carreteras.
- W: arreglo de longitud $N-1$ que describe las longitudes de las carreteras.
- Esta función debe retornar la máxima puntuación de conveniencia que se puede conseguir por alguna asignación de horas de cierre.
- La función puede llamarse **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])
```

Esta corresponde a la siguiente red de carreteras:



Supón que las horas de cierre se asignan de la siguiente forma:

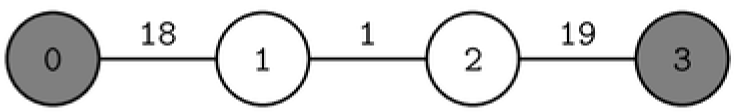
Ciudad	0	1	2	3	4	5	6
Hora de cierre	0	4	0	3	2	0	0

Nota que la suma de todas las horas de cierre es 9, que no es mayor 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 tanto, la puntuación de conveniencia es $3 + 3 = 6$. No existe ninguna otra asignación de horas de cierre con puntuación de conveniencia mayor a 6, así que la función debe retornar 6.

También considera la siguiente llamada:

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

Esta corresponde a la siguiente rede de carreteras:



Supón que las horas de cierre se asignan de la siguiente forma:

Ciudad	0	1	2	3
Hora 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 tanto, la puntuación de conveniencia es $1 + 2 = 3$. No hay asignación de horas de cierre con puntuación de conveniencia mayor a 3, así que la función debe retornar 3.

Restricciones

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

Sub-tareas

Decimos que una red de carreteras es **lineal** si la carretera i conecta las ciudades i y $i+1$ (para cada i tal que $0 \leq i \leq N-2$).

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

Evaluador de ejemplo

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

- línea 1: C

Luego siguen las descripciones de los C escenarios.

El grader de ejemplo lee la descripción de cada escenario en el siguiente formato:

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

El grader de ejemplo imprime una sola línea por cada escenario en el siguiente formato:

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