Tiempos de Cierre

Hungría es un país con N ciudades, numeradas del 0 al $N-1$.

Las ciudades están conectadas por $N-1$ carreteras *bidireccionales*, numeradas del 0 al $N-2$. Para toda j tal que $0 \le j \le N-2$, la carretera j une a la ciudad $U[j]$ y a la ciudad $V[j]$ y tiene longitud $W[j]$, es decir, permite viajar entre estas ciudades en $W[j]$ unidades de tiempo. Cada carretera conecta a dos ciudades distintas y cada par de ciudades tiene a lo más una carretera entre ellas.

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 toda i ($0 \le i < t$), existe una carretera que conecta a las ciudades p_i y p_{i+1}.

Es posible viajar entre cualesquiera dos ciudades utilizando las carreteras, es decir, existe un camino entre cualesquiera dos ciudades distintas. Es posible demostrar que este camino es único para cualquier 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 a las ciudades consecutivas del camino.

En Hungría, muchas personas viajan para visitar las festividades del Dia de la Fundación en dos de las ciudades principales. Una vez que las festividades terminan, estas personas regresan a sus hogares. El gobierno quiere prevenir que la multitud perturbe a los locales, entonces planean cerrar todas las ciudades a ciertas horas. Cada ciudad tendrá un **tiempo de cierre** no negativo asignado por el gobierno. El gobierno ha decidido que la suma de todos los tiempos de cierre no debe ser mayor a K. Más precisamente, para toda i entre 0 y $N-1$ inclusivo, el tiempo de cierre de la ciudad i es un entero no negativo $c[i]$. La suma de todos los $c[i]$ no debe ser mayor a K.

Considera una ciudad a y alguna asignación de tiempos de cierre. Decimos que la ciudad b es **alcanzable** desde la ciudad a si y solo si, o $b = a$ 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 más $c[p_1]$ y
- la longitud del camino p_0, p_1, p_2 es a lo más $c[p_2]$ y
- $\ldots$

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

Este año, las dos ciudades principales de las festividades están ubicadas en la ciudad X y en la ciudad Y. Para cada asignación de tiempos de cierre, el **puntaje de conveniencia** se define como la suma de:

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

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

Tu tarea es encontrar el máximo puntaje de conveniencia que alguna asignación de tiempos de cierre puede tener.

Detalles de Implementación

Deberás implementar la siguiente función:

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

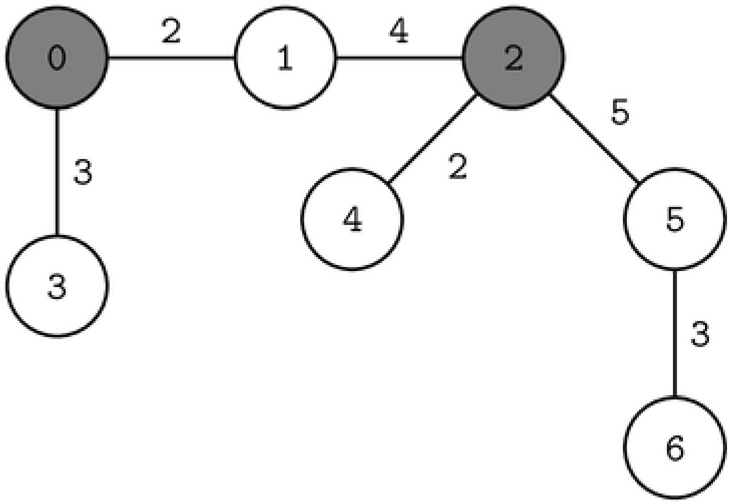
- N: el número de ciudades.
- X, Y: las dos ciudades principales de las festividades.
- K: el máximo valor de la suma de los tiempos de cierre.
- U, V: arreglos de longitud $N - 1$ que describen las carreteras.
- W: arreglo de longitud $N - 1$ que describe la longitud de las carreteras.
- Esta función debera regresar el máximo puntaje de conveniencia que alguna asignación de tiempos de cierre puede tener.
- Esta función puede ser llamada **multiples veces** en cada caso de prueba.

Ejemplo

Considera la siguiente llamada a la función:

```
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 a la siguiente red de carreteras:



Supón que los tiempos de cierre se asignan así:

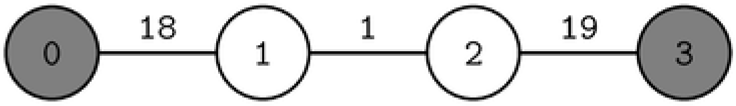
Ciudad	0	1	2	3	4	5	6
Tiempo de cierre	0	4	0	3	2	0	0

Nota que la suma de todos los tiempos de cierre es 9, la cual no es mayor que $K = 10$. Las ciudades 0, 1 y 3 son alcanzables desde la ciudad X (X = 0), y las ciudades 1, 2 y 4 son alcanzables desde la ciudad Y ($Y = 2$). Por lo tanto el puntaje de conveniencia es $3 + 3 = 6$. No existe alguna asignación de tiempos de cierre con un puntaje de conveniencia mayor a 6, entonces la función debe regresar 6.

Ahora considera la siguiente llamada a la función:

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

Esto corresponde a la siguiente red de carreteras:



Supón que los tiempos de cierre se asignan así:

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

La ciudad 0 es alcanzable desde la ciudad X (X = 0), y las ciudades 2 y 3 son alcanzables desde la ciudad Y ($Y = 3$). Por lo tanto el puntaje de conveniencia es $1 + 2 = 3$. No existe alguna asignación de tiempos de cierre con un puntaje de conveniencia mayor a 3, entonces la función debe regresar 3.

Límites

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

Subtareas

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

1. (8 puntos) La longitud del camino de la ciudad X a la ciudad Y es mayor a $2K$.
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 casos, es decir, el número de llamadas a `max_score`. El evaluador de ejemplo lee la entrada en el siguiente formato:

- línea 1: C

Seguido de las descripciones de los C casos.

El evaluador de ejemplo lee la descripción de cada caso 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 evaluador de ejemplo imprime una única linea para cada caso, en el siguiente formato:

- línea 1: el valor regresado por `max_score`