Horas 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 $0 \le j \le N-2$, la carretera j conecta la ciudad $U[j]$ y la ciudad $V[j]$ y tiene una longitud $W[j]$, es decir, permite viajar entre las ciudades en $W[j]$ unidades de tiempo. Cada carretera conecta dos ciudades diferentes, y cada par de ciudades está conectado como máximo por una carretera.

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

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

Es posible viajar de cualquier ciudad a cualquier otra utilizando las carreteras, es decir, existe un camino entre cada dos ciudades distintas. Se puede demostrar 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 los caminos t 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 algunas de las principales ciudades. Una vez terminadas las celebraciones, regresan a sus hogares. El gobierno quiere evitar que la multitud moleste a los lugareños, por lo que planea cerrar las ciudades a determinadas horas. El gobierno asignará a cada ciudad una **hora de cierre** no negativa. El gobierno ha decidido que la suma de todas las horas de cierre no debe ser superior a K. Más concretamente, para cada i entre 0 y $N-1$, ambos inclusive, la hora de cierre asignada a la ciudad i es un número entero no negativo $c[i]$. La suma de todos los $c[i]$ no debe ser mayor que K.

Consideremos una ciudad a y algunas asignaciones de horas de cierre. Decimos que una ciudad b es **alcanzable** desde la ciudad a si y sólo si o bien $b = a$, o bien el camino $p_0, \ldots, p_t$ entre estas dos ciudades (por lo que en particular $p_0 = a$ y $p_t = b$) satisface las siguientes condiciones:

- la longitud del camino p_0, p_1 es como máximo $c[p_1]$, y
- la longitud del camino p_0, p_1, p_2 es como máximo $c[p_2]$, y

- ...
- la longitud del camino $p_0, p_1, p_2, \ldots, p_t$ es como máximo $c[p_t]$.

Este año, las dos sedes principales del festival están situadas en la ciudad X y en la ciudad Y. Para cada asignación de horarios de cierre, la **puntuación de conveniencia** se define como la suma de los dos números siguientes:

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

Tenga en cuenta que si se puede llegar a una ciudad desde la ciudad X y desde la ciudad Y, cuenta *dos veces* para la puntuación de conveniencia.

Su tarea consiste en calcular la máxima puntuación de conveniencia que se puede conseguir con una cierta asignación de horas de cierre.

Detalles de la implementación

Usted debe implementar el siguiente procedimiento.

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

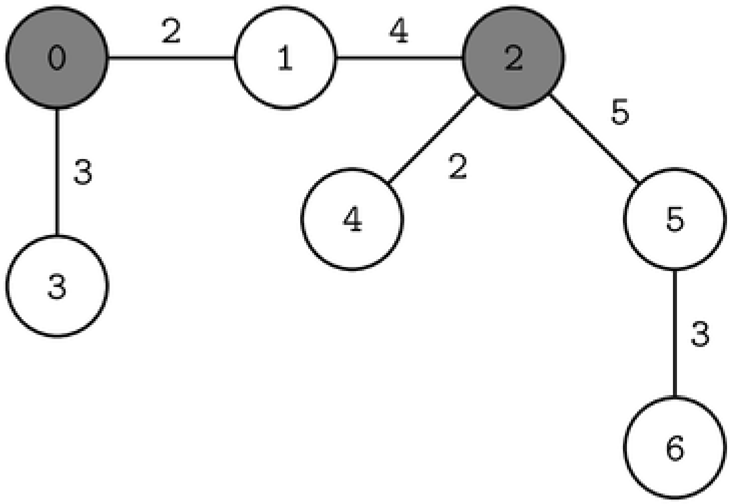
- N: el número de ciudades.
- X, Y: las ciudades con sedes principales de festivales.
- K: el límite superior de la suma de horas de cierre.
- U, V: matrices de longitud $N-1$ que describen las conexiones por carretera.
- W: matriz de longitud $N-1$ que describe la longitud de las carreteras.
- Este procedimiento debe devolver la máxima puntuación de conveniencia que puede lograrse mediante una asignación de horas de cierre.
- Este procedimiento puede llamarse **múltiples veces** en cada caso de prueba.

Ejemplo

Considere 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 a la siguiente red de carreteras:



Supongamos que las horas de cierre se asignan de la siguiente manera:

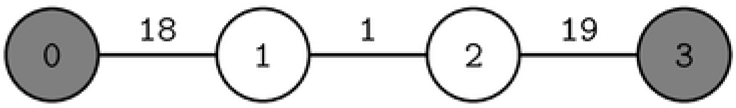
Ciudad	0	1	2	3	4	5	6
Hora de cierre	0	4	0	3	2	0	0

Obsérvese que la suma de todas las horas de cierre es 9, que no es más que $K = 10$. Las ciudades 0, 1 y 3 son accesibles desde la ciudad X ($X = 0$), mientras que las ciudades 1, 2 y 4 son accesibles desde la ciudad Y ($Y = 2$). Por lo tanto, la puntuación de conveniencia es $3 + 3 = 6$. No hay ninguna asignación de horas de cierre con una puntuación de conveniencia superior a 6, por lo que el procedimiento debe devolver 6.

Considera también 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 carreteras:



Supongamos que las horas de cierre se asignan de la siguiente manera:

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

A la ciudad 0 se puede llegar desde la ciudad X ($X = 0$), mientras que las ciudades 2 y 3 se puede llegar desde la ciudad Y ($Y = 3$). Por lo tanto, la puntuación de conveniencia es $1 + 2 = 3$. No hay ninguna asignación de horas de cierre con una puntuación de conveniencia superior a 3, por lo que el procedimiento debe devolver 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.

Subtareas

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 de 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 viaria 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.

Calificador de ejemplo Grader

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

- línea 1: C

Las descripciones de C escenarios siguen.

El calificador 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 calificador de ejemplo imprime una sola línea para cada escenario, con el siguiente formato:

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