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 la ciudad $U[j]$ y la ciudad $V[j]$ y tiene longitud $W[j]$, o sea, permite viajar entre las ciudades en $W[j]$ unidades de tiempo. Cada carretera conecta dos ciudades distintas, y cada par de ciudades tiene como mucho 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 ($0 \leq i < t$), hay una carretera entre p_i y p_{i+1}.

Es posible viajar desde cualquier ciudad a cualquier otra, es decir, existe un camino entre cualquier par de ciudades distintas. Se puede demostrar que en estas condiciones este camino es único.

La **longitud** de un camino $p_0, p_1, \ldots, p_t$ es la suma de las longitudes de las t carreteras conectando ciudades consecutivas del camino.

En Hungría, mucha gente viaja para participar en las festividades del Día de la Fundación en dos de las ciudades más importantes. Una vez han acabado las festividades, vuelven a sus hogares. El gobierno quiere prevenir que las multitudes perturben el descanso de la gente, así que planean cerrar todas las ciudades del país a unas horas determinadas. A cada una de las ciudades le será asignada una **hora de cierre** (no negativa) por el gobierno. El gobierno ha decidido que la suma de las horas de cierre no debe ser mayor que K. Formalmente, para cada i entre 0 y $N-1$, inclusive, la hore de cierre assignada a la ciudad i es un entero no negativo $c[i]$. La suma de todos los $c[i]$ no debe ser mayor que K.

Consideremos una ciudad a y una asignación 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 (que en particular satisface $p_0 = a$ y $p_t = b$) cumple las siguientes condiciones:

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

Este año, las festividades están localizadas en la ciudad X y la ciudad Y. Para cada asignación de horas 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.

Nótese que, si una ciudad es alcanzable desde tando la ciudad X como la Y, se cuenta *por duplicado* para calcular la puntuación de conveniencia.

Tu tarea es calcular la máxima puntuación de conveniencia que puede ser obtenida por una 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)
```

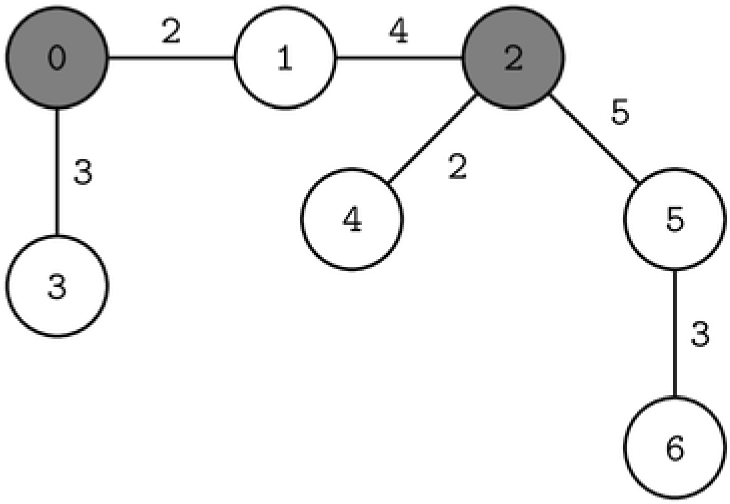
- N: el número de ciudades.
- X, Y: las ciudades en las que se celebran las festividades.
- K: la cota superior a la suma de horas de cierre.
- U, V: arrays de longitud $N-1$ describiendo las conexiones de las carreteras.
- W: array de longitud $N-1$ describiendo las longitudes de las carreteras.
- Esta función debe retornar la máxima puntuación de conveniencia que puede ser obtenida por alguna asignación de horas de cierre.
- Esta función puede ser llamada **varias 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])
```

Corresponde a la siguiente disposición de carreteras:



Supóngase que las horas de cierre son asignadas de la siguiente forma:

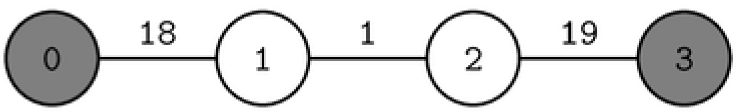
Ciudad	0	1	2	3	4	5	6
Hora de cierre	0	4	0	3	2	0	0

Nótese que la suma de todas las horas de cierre es 9, que es menor o igual 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 tanto, la puntuación de conveniencia es $3 + 3 = 6$. No existe ninguna asignación de horas de cierre con puntuación de conveniencia mayor que 6, así que la función debe retornar 6.

Además, considera la siguiente llamada:

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

Corresponde a la siguiente disposición de carreteras:



Supóngase que las horas de cierre son asignadas 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 ($X = 0$), y las ciudades 2 y 3 son alcanzables desde la ciudad Y ($Y = 3$). Por tanto, la puntuación de conveniencia es $1 + 2 = 3$. No existe ninguna asignación de horas de cierre con puntuación de conveniencia mayor que 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 usando las carreteras.
- $S_N \leq 200\,000$, donde S_N es la suma de N sobre todas las llamadas a `max_score` en un caso de prueba.

Subtareas

Decimos que una disposición 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 disposición de carreteras es lineal.
3. (12 puntos) $S_N \leq 500$, la disposición de carreteras es lineal.
4. (14 puntos) $S_N \leq 3\,000$, la disposición 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.

Grader de ejemplo

Sea C el número de situaciones, es decir, el número de llamadas a `max_score`. El grader de ejemplo lee la entrada con el siguiente formato:

- línea 1: C

Las descripciones de C situaciones siguen a continuación.

El grader de ejemplo lee la descripción de cada situación con 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 línea por cada situación, con el siguiente formato:

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