Tempo de Encerramento

A Hungria é um país com N cidades, numeradas de 0 a $N-1$.

As cidades são conectadas por $N-1$ estradas *bidirecionais*, numeradas de 0 a $N-2$. Para cada j tal que $0 \leq j \leq N-2$, a estrada j conecta a cidade $U[j]$ e a cidade $V[j]$ e possui tamanho $W[j]$, isto é, ela permite viajar entre as cidades em $W[j]$ unidades de tempo. Cada estrada conecta duas cidades diferentes, e cada par de cidades é conectado por no máximo uma estrada.

Um **caminho** entre duas cidades distintas a e b é uma sequência $p_0, p_1, \ldots, p_t$ de cidades distintas, tal que:

- $p_0 = a$,
- $p_t = b$,
- para cada i ($0 \leq i < t$), existe uma estrada conectando as cidades p_i e p_{i+1}.

É possível viajar de qualquer cidade para qualquer outra cidade usando as estradas, isto é, existe um caminho entre duas quaisquer cidades distintas. Note que esse caminho é único para cada par de cidades distintas.

O **tamanho** do caminho $p_0, p_1, \ldots, p_t$ é a soma dos tamanhos das t estradas conectando as cidades consecutivas ao longo do caminho.

Na Hungria, muitas pessoas viajam para participar dos festivais do Dia da Fundação em algumas das maiores cidades. Quando a celebração terminar, eles retornarão para suas casas. O governo quer prevenir que a multidão perturbe os cidadãos locais, e planejam então bloquear as cidades em certos tempos. Para cada cidade é designado um **tempo de fechamento** não negativo pelo governo. O governo decidiu que a soma de todos os *tempos de fechamento* não deve ser maior que K. Mais precisamente, para cada i entre 0 e $N-1$, inclusive, o tempo de fechamento designado para a cidade i é um inteiro não negativo $c[i]$ A soma de todos os $c[i]$ não pode ser maior que K.

Considere uma cidade a e alguma escolha de *tempos de fechamento*. Dizemos que a cidade b é **alcançável** a partir da cidade a se e somente se $b = a$ ou o caminho $p_0, \ldots, p_t$ entre estas duas cidades (então, em particular, $p_0 = a$ e $p_t = b$) satisfaz as seguintes condições:

- o tamanho do caminho p_0, p_1 é no máximo $c[p_1]$, e
- o tamanho do caminho p_0, p_1, p_2 é no máximo $c[p_2]$, e
- $\ldots$
- o tamanho do caminho $p_0, p_1, p_2, \ldots, p_t$ é no máximo $c[p_t]$.

Nesse ano, os dois principais locais dos festivais estão localizados na cidade X e na cidade Y. Para cada escolha de *tempos de fechamento*, a **pontuação de conveniência** é definida como soma dos dois seguintes números:

- O número de cidades alcançáveis a partir da cidade X.
- O número de cidades alcançáveis a partir da cidade Y.

Note que se uma cidade é alcançável a partir de ambas as cidades X e Y, ela é contabilizada *duas vezes* na *pontuação de conveniência*.

Sua tarefa é computar a máxima *pontuação de conveniência* que pode ser alcançada por alguma escolha de *tempos de fechamento*.

Detalhes de Implementação

Você deve implementar o seguinte procedimento.

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

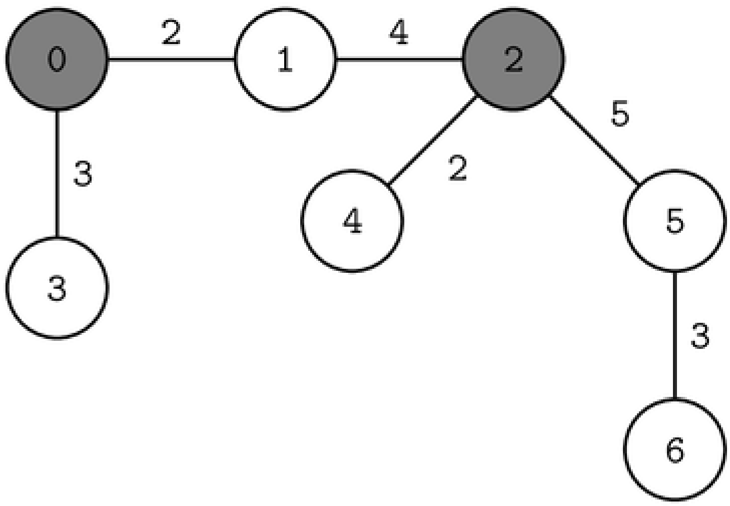
- N: o número de cidades.
- X, Y: as cidades com os principais locais dos festivais.
- K: o limite da soma dos *tempo de fechamento*.
- U, V: vetores de tamanho $N-1$ descrevendo as conexões das estradas.
- W: vetor de tamanho $N-1$ descrevendo os tamanhos das estradas.
- Este procedimento deve retornar a máxima *pontuação de conveniência* que pode ser alcançada por uma escolha de *tempos de fechamento*.
- Este procedimento pode ser chamado **múltiplas vezes** em cada caso de teste.

Exemplo

Considere a seguinte chamada:

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

Isso corresponde à seguinte rede de estradas:



Suponha que os *tempos de fechamento* são escolhidos da seguinte forma:

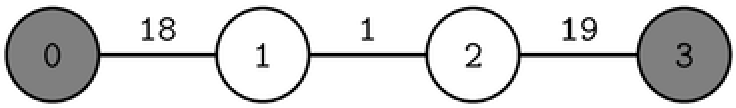
Cidade	0	1	2	3	4	5	6
Tempo de fechamento	0	4	0	3	2	0	0

Note que a soma dos *tempos de fechamento* é 9, que não é maior que $K = 10$. As cidades 0, 1, e 3 são alcançávies a partir da cidade X ($X = 0$), enquanto as cidades 1, 2, e 4 são alcançávies a partir da cidade Y ($Y = 2$). Logo, a *pontuação de conveniência* é $3 + 3 = 6$. Não existe nenhuma escolha de *tempos de encerramento* com *pontuação de conveniência* maior que 6, então o procedimento deve retornar 6.

Considere também a seguinte chamada:

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

Isso corresonde à seguinte rede de estradas:



Suponha que os *tempos de fechamento* são escolhidos da seguinte forma:

Cidade	0	1	2	3
Tempo de fechamento	0	1	19	0

A cidade 0 é alcançável a partir da cidade X ($X = 0$), enquanto as cidades 2 e 3 são alcançávies a partir da cidade Y ($Y = 3$). Logo, a *pontuação de conveniência* é $1 + 2 = 3$. Não existe nenhuma escolha de *tempos de encerramento* com *pontuação de conveniência* maior que 3, então o procedimento deve retornar 3.

Restrições

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (para cada j tal quet $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (para cada j tal que $0 \leq j \leq N - 2$)
- É possível viajar de qualquer cidade para qualquer outra cidade usando apenas as estradas.
- $S_N \leq 200\,000$, onde S_N é a soma de N em todas as chamadas de `max_score`.

Subtarefas

Dizemos que uma rede de estradas é **linear** se a estrada i conecta as cidades i e $i + 1$ (para cada i tal que $0 \leq i \leq N - 2$).

1. (8 pontos) O tamanho do caminho da cidade X para a cidade Y é maior que $2K$.
2. (9 pontos) $S_N \leq 50$, a rede de estradas é linear.
3. (12 pontos) $S_N \leq 500$, a rede de estradas é linear.
4. (14 pontos) $S_N \leq 3\,000$, a rede de estradas é linear.
5. (9 pontos) $S_N \leq 20$
6. (11 pontos) $S_N \leq 100$
7. (10 pontos) $S_N \leq 500$
8. (10 pontos) $S_N \leq 3\,000$
9. (17 pontos) Sem restrições adicionais.

Corretor Exemplo

Seja C o número de cenários, isto é, o número de chamadas para `max_score`. O corretor exemplo lê a entrada no seguinte formato:

- linha 1: C

Seguem as descrições dos C cenários.

O corretor exemplo lê a descrição de cada cenário no seguinte formato:

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

O corretor exemplo imprime uma única linha para cada cenário, no seguinte formato:

- linha 1: o valor de retorno de `max_score`