# Closing Time

Венгрия состоит из $N$ городов, пронумерованных от $0$ до $N-1$.

Города соединены $N-1$ *двусторонними* дорогами, дороги пронумерованы от $0$ до $N-2$. Для всех $j$, таких что $0 \le j \le N-2$, дорога $j$ соединяет города $U[j]$ и $V[j]$ и имеет длину $W[j]$, эта дорога позволяет проехать из одного города в другой за $W[j]$ единиц времени. Каждая дорога соединяет два различных города, любые два города соединены не более одной дорогой.

**Путем** между двумя различными городами $a$ и $b$ называется последовательность различных городов $p_0, p_1, \ldots, p_t$, такая что:

- $p_0 = a$,
- $p_t = b$,
- для всех $i$ ($0 \le i < t$) есть дорога, соединяющая города $p_i$ и $p_{i+1}$.

Из любого города можно доехать до любого другого используя дороги, то есть существует путь между любыми двумя различными городами. Можно показать, что этот путь единственный для каждой пары различных городов.

**Длиной** пути $p_0, p_1, \ldots, p_t$ называется сумма длин $t$ дорог, соединяющих соседние города вдоль пути.

В Венгрии многие люди ездят на фестиваль в честь дня города в два крупных города. Когда фестиваль заканчивается, они возвращаются домой. Правительство хочет, чтобы толпа людей не помешала местным жителям, поэтому они планируют закрыть города в некоторые моменты времени. Правительство назначит каждому городу неотрицательное **время закрытия**. Правительство решило, что сумма времен закрытия должна быть не больше $K$. Более формально, для любого $i$ $(0 \le i \le N-1)$ время закрытия города $i$ это неотрицательное целое число $c[i]$. Сумма $c[i]$ должна быть не больше $K$.

Рассмотрим город $a$ и некоторое распределение времен закрытия. Мы говорим, что город $b$ **достижим** из города $a$ тогда и только тогда, когда $b = a$, или путь $p_0, \ldots, p_t$ между этими двумя городами (то есть $p_0 = a$ и $p_t = b$) удовлетворяет следующим условиям:

- длина пути $p_0, p_1$ не превосходит $c[p_1]$, и
- длина пути $p_0, p_1, p_2$ не превосходит $c[p_2]$, и
- $\ldots$

- длина пути $p_0, p_1, p_2, \ldots, p_t$ не превосходит $c[p_t]$.

В этом году два главных фестиваля пройдут в городах $X$ и $Y$. Для каждого распределения времен закрытия назовем **оценкой удобства** сумму следующих двух чисел:

- Количество городов достижимых из города $X$.
- Количество городов достижимых из города $Y$.

Обратите внимание, что если город достижим из города $X$ и достижим из города $Y$, он считается *дважды* в оценке удобства.

Найдите максимальную оценку удобства, которую можно получить для некоторого распределения времен закрытия.

# Implementation Details

Вы должны реализовать следующую функцию.

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

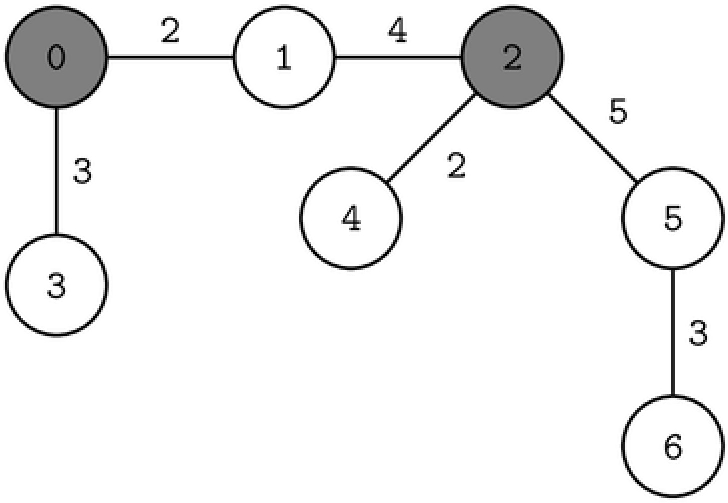
- $N$: количество городов.
- $X$, $Y$: города, в которых пройдут главные фестивали.
- $K$: ограничение сверху на сумму времен закрытия.
- $U$, $V$: массивы длины $N - 1$ описывающие дороги.
- $W$: массив длины $N - 1$ описывающий длины дорог.
- Эта функция должна вернуть максимальную оценку удобства, которая может быть получена с помощью некоторого распределения времен закрытия.
- Эта функция может быть вызвана **несколько раз** в одном тесте.

# Example

Рассмотрим следующий вызов функции:

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

Он соответствует следующей дорожной сети:




Допустим времена закрытия распределены следующим образом:

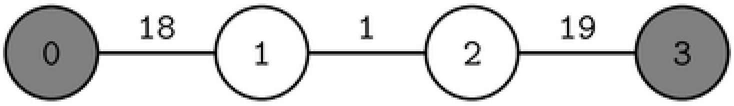
| **Город** | $0$ | $1$ | $2$ | $3$ | $4$ | $5$ | $6$ |
|---|---|---|---|---|---|---|---|
| **Время закрытия** | $0$ | $4$ | $0$ | $3$ | $2$ | $0$ | $0$ |

Обратите внимание, что сумма времен закрытия равна $9$, что не превосходит $K = 10$. Города $0$, $1$ и $3$ достижимы из города $X$ ($X = 0$), тогда как города $1$, $2$ и $4$ достижимы из города $Y$ ($Y = 2$). Таким образом, оценка удобства равна $3 + 3 = 6$. Не существует ни одного распределения времен закрытия, которое имеет оценку удобства больше $6$, поэтому функция должна вернуть $6$.

Также рассмотрим следующий вызов функции:

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

Он соответствует следующей дорожной сети:




Допустим времена закрытия распределены следующим образом:

| **Город** | $0$ | $1$ | $2$ | $3$ |
|---|---|---|---|---|
| **Время закрытия** | $0$ | $1$ | $19$ | $0$ |

Город $0$ достижим из города $X$ ($X = 0$), тогда как города $2$ и $3$ достижимы из города $Y$ ($Y = 3$). Таким образом, оценка удобства равна $1 + 2 = 3$. Не существует ни одного распределения времен закрытия, которое имеет оценку удобства больше $3$, поэтому функция должна вернуть $3$.

## Constraints

- $2 \le N \le 200\,000$
- $0 \le X < Y < N$
- $0 \le K \le 10^{18}$
- $0 \le U[j] < V[j] < N$ (для всех $j$ таких что $0 \le j \le N - 2$)
- $1 \le W[j] \le 10^6$ (для всех $j$ таких что $0 \le j \le N - 2$)
- От любого города можно доехать до любого другого, используя дороги.
- $S_N \le 200\,000$, где $S_N$ это сумма $N$ по всем вызовам функции `max_score` в одном тесте.

## Subtasks

Будем называть дорожную сеть **линейной**, если дорога $i$ соединяет города $i$ и $i + 1$ (для всех $i$ таких что $0 \le i \le N - 2$).

1. (8 баллов) Длина пути между городом $X$ и городом $Y$ больше чем $2K$.
2. (9 баллов) $S_N \le 50$, дорожная сеть линейная.
3. (12 баллов) $S_N \le 500$, дорожная сеть линейная.
4. (14 баллов) $S_N \le 3\,000$, дорожная сеть линейная.
5. (9 баллов) $S_N \le 20$
6. (11 баллов) $S_N \le 100$
7. (10 баллов) $S_N \le 500$
8. (10 баллов) $S_N \le 3\,000$
9. (17 баллов) Нет дополнительных ограничений.

## Sample Grader

Пусть $C$ обозначает число сценариев в тесте, иначе говоря, число вызовов функции `max_score`. Грейдер читает тест в следующем формате:

- строка 1: $C$

Далее следуеют описания $C$ сценариев в следующем формате:

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

Грейдер выводит единственную строку для каждого сценария в следующем формате:

- строка 1: значение, которое вернула функция `max_score`