



زمان بستن

کشور مجارستان شامل N شهر است که از 0 تا $N - 1$ شماره‌گذاری شده‌اند.

این شهرها با $N - 1$ جاده دوطرفه با شماره‌های 0 تا $N - 2$ به هم متصل‌اند. به ازای هر j با شرط $0 \leq j \leq N - 2$ ، جاده j شهرهای $U[j]$ و $V[j]$ را به هم وصل می‌کند و دارای طول $W[j]$ است، یعنی با استفاده از این جاده می‌توان بین دو شهر در $W[j]$ واحد زمان سفر کرد. هر جاده دو شهر متفاوت را به هم وصل می‌کند و هر دو شهر متمایز با حداکثر یک جاده به هم متصل شده‌اند.

یک مسیر بین دو شهر متمایز a و b یک دنباله p_0, p_1, \dots, p_t از شهرهای متمایز است، طوری که:

- $p_0 = a$
- $p_t = b$
- به ازای هر i ($0 \leq i < t$)، یک جاده بین p_i و p_{i+1} وجود دارد.

با استفاده از جاده‌ها می‌توان از هر شهر به هر شهر دیگر سفر کرد، یعنی بین هر دو شهر متمایز یک مسیر وجود دارد. دقت کنید که این مسیر به ازای هر دو شهر متمایز یکتا است.

طول یک مسیر p_0, p_1, \dots, p_t برابر مجموع طول t جاده‌ای است که شهرهای متوالی را در طول مسیر به هم متصل می‌کند.

در مجارستان، جمعیت زیادی برای شرکت در جشن‌های روز پایه‌گذاری در برخی شهرهای مهم شرکت می‌کنند. وقتی جشن‌ها خاتمه می‌یابند، مردم به خانه‌های خود برمی‌گردند. برای این که جمعیت باعث ایجاد مزاحمت برای ساکنین شهرها نشود، دولت می‌خواهد شهرها را در زمان‌های خاصی تعطیل کند. در این راستا به هر شهر یک عدد نامنفی به عنوان زمان بستن توسط دولت اختصاص می‌یابد. دولت تصمیم گرفته است که مجموع تمام زمان‌های بستن نباید از یک عدد K بیشتر شود. به طور دقیق‌تر، به ازای هر i بین 0 و $N - 1$ ، زمان بستن اختصاص یافته به شهر i برابر عدد نامنفی $c[i]$ است. مجموع تمام $c[i]$ ها نباید از K بیشتر شود.

یک شهر a را به همراه تخصیصی از زمان‌های بستن به شهرها در نظر بگیرید. می‌گوییم شهر a از شهر b قابل رسیدن است اگر و فقط اگر یا $b = a$ ، یا مسیری مانند p_0, \dots, p_t بین این دو شهر وجود داشته باشد ($p_0 = a$ و $p_t = b$) که در شرایط زیر صدق کند:

- طول مسیر p_0, p_1 حداکثر $c[p_1]$ باشد، و
- طول مسیر p_0, p_1, p_2 حداکثر $c[p_2]$ باشد، و
- ...
- طول مسیر $p_0, p_1, p_2, \dots, p_t$ حداکثر $c[p_t]$ باشد.

امسال، جشن‌های اصلی در دو شهر X و Y برگزار می‌شوند. به ازای هر تخصیص ممکن از زمان‌های بستن، امتیاز آسودگی برابر مجموع دو عدد زیر تعریف می‌شود:

- تعداد شهرهای قابل رسیدن از X .
- تعداد شهرهای قابل رسیدن از Y .

دقت کنید که اگر یک شهر هم از شهر X و هم از شهر Y قابل رسیدن باشد، دو بار در امتیاز آسودگی محاسبه می‌شود.

کار شما این است که بیشترین زمان آسودگی که می‌توان با تخصیصی از زمان‌های بستن به آن رسید محاسبه کنید.

Implementation Details

You should implement the following procedure

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

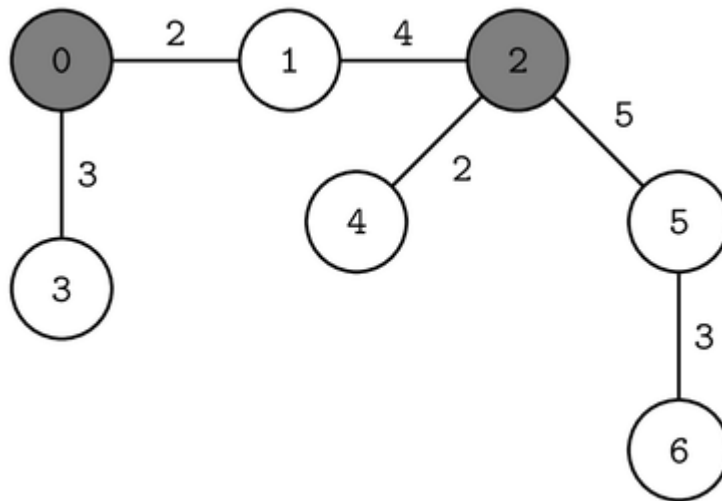
- N : the number of cities
- X, Y : the cities with main festival sites
- K : the upper bound on the sum of closing times
- U, V : arrays of length $N - 1$ describing road connections
- W : array of length $N - 1$ describing road lengths
- This procedure should return the maximum convenience score that can be achieved by some assignment of closing times
- This procedure may be called **multiple times** in each test case

Example

Consider the following call

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

This corresponds to the following road network



:Suppose the closing times are assigned as follows

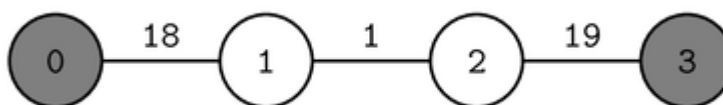
City	0	1	2	3	4	5	6
Closing time	0	4	0	3	2	0	0

Note that the sum of all closing times is 9, which is not more than $K = 10$. Cities 0, 1, and 3 are reachable from city X ($X = 0$), while cities 1, 2, and 4 are reachable from city Y ($Y = 2$). Therefore, the convenience score is $3 + 3 = 6$. There is no assignment of closing times with .6 convenience score more than 6, so the procedure should return

:Also consider the following call

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

:This corresponds to the following road network



:Suppose the closing times are assigned as follows

City	0	1	2	3
Closing time	0	1	19	0

City 0 is reachable from city X ($X = 0$), while cities 2 and 3 are reachable from city Y ($Y = 3$). Therefore, the convenience score is $1 + 2 = 3$. There is no assignment of closing times with

.3 convenience score more than 3, so the procedure should return

Constraints

- $2 \leq N \leq 200\,000$ •
- $0 \leq X < Y < N$ •
- $0 \leq K \leq 10^{18}$ •
- $(0 \leq j \leq N - 2 \text{ for each } j \text{ such that}) 0 \leq U[j] < V[j] < N$ •
- $(0 \leq j \leq N - 2 \text{ for each } j \text{ such that}) 1 \leq W[j] \leq 10^6$ •
- .It is possible to travel from any city to any other city by using the roads •
- .where S_N is the sum of N over all calls to `max_score` , $S_N \leq 200\,000$ •

Subtasks

We say that a road network is **linear** if road i connects cities i and $i + 1$ (for each i such that $0 \leq i \leq N - 2$).

- .2K points) The length of the path from city X to city Y is greater than 8) .1
- .points) $S_N \leq 50$, the road network is linear 9) .2
- .points) $S_N \leq 500$, the road network is linear 12) .3
- .points) $S_N \leq 3\,000$, the road network is linear 14) .4
- $S_N \leq 20$ (points 9) .5
- $S_N \leq 100$ (points 11) .6
- $S_N \leq 500$ (points 10) .7
- $S_N \leq 3\,000$ (points 10) .8
- .points) No additional constraints 17) .9

Sample Grader

Let C denote the number of scenarios, that is, the number of calls to `max_score`. The sample grader reads the input in the following format

C :1 line •

.The descriptions of C scenarios follow

:The sample grader reads the description of each scenario in the following format

$N \ X \ Y \ K$:1 line •

$U[j] \ V[j] \ W[j]$: $(0 \leq j \leq N - 2) \ 2 + j$ line •

:The sample grader prints a single line for each scenario, in the following format

line 1: the return value of `max_score` •