



## Closing Time

Венгрияда 0-ден  $N - 1$ -ге дейін нөмірленген  $N$  қала бар.

Қалалар 0-ден  $N - 2$ -ге дейін нөмірленген  $N - 1$  *екі жақты* тікелей жолмен байланысқан. Әрбір  $j$  ( $0 \leq j \leq N - 2$ ) үшін,  $j$  тікелей жолы  $U[j]$  қаланы және  $V[j]$  қаланы байланыстырады және ұзындығы  $W[j]$  болады, яғни, қалалар арасында  $W[j]$  уақыт бірлігінде жүруге мүмкіндік береді. Әр бір тікелей жол екі әр түрлі қаланы байланыстырады, және қала жұбы ең көбі бір тікелей жол арқылы байланысады.

Екі әр түрлі  $a$  және  $b$  қаланын арасындағы **жол** деп  $p_0, p_1, \dots, p_t$  әр түрлі қалалар тізбегін айтамыз:

- $p_0 = a$ ,
- $p_t = b$ ,
- әр бір  $i$  ( $0 \leq i < t$ ) үшін,  $p_i$  және  $p_{i+1}$  қалаларын байланыстыратын тікелей жол бар.

Тікелей жолдар арқылы кез-келген қаладан кез-келген басқа қалаға баруға болады, яғни, кез келген екі әр түрлі қалалардың арасында жол бар. Әр қала жұбы үшін бұл жолдың жалғыз ғана екенін көрсетуге болады.

$p_0, p_1, \dots, p_t$  жолдың **ұзындығы** қалаларды байланыстыратын  $t$  тікелей жолдардың ұзындықтарының қосынды болып табылады.

Венгрияда көп адамадар Құрастыру Күнін көруге екі әр түрлі фестиваль алаңы бар қалаға келеді. Мереке аяқталғаннан кейін олар үйлеріне қайтады. Үкімет жергілікті тұрғындарды мазаламауын қалайды, сондықтан олар барлық қалаларды белгілі бір уақытқа жабуды жоспарлап отыр. Үкімет әр қалаға теріс емес **жабылу уақытын** тағайындайды. Үкімет барлық жабылу уақыттардың қосындысы  $K$ -ны аспауы керек деп шешті. Дәлірек айтқанда, 0 мен  $N - 1$  арасындағы әрбір  $i$  үшін,  $c[i]$  жабылу уақыты  $i$  қаласына тағайындалады. Барлық  $c[i]$  қосындысы  $K$ -дан көп емес болуы керек.

$a$  қаласын және кейбір жабылу уақытырын қарастырайық. Біз  $b$  қаласын  $a$  қаласының **жетімді** деп айтамыз сонда және тек сонда ғана, қашан  $b = a$ , немесе  $p_0, \dots, p_t$  екі қала арасындағы жол (яғни  $p_0 = a$  және  $p_t = b$ ) келесі шарттарды қанағаттандырады:

- $p_0, p_1$  жолдың ұзындығы  $c[p_1]$ -ден көп емес
- $p_0, p_1, p_2$  жолдың ұзындығы  $c[p_2]$ -ден көп емес
- ...

- $p_0, p_1, p_2, \dots, p_t$  жолдын ұзындығы  $c[p_t]$ -ден көп емес

Биыл екі негізгі фестиваль алаңы  $X$  және  $Y$  қалаларында орналасқан. Әрбір жабылу уақыты үшін **ыңғайлылық** келесі екі санның қосындысы ретінде анықталады:

- $X$  қаласынан жетімді қалалар саны.
- $Y$  қаласынан жетімді қалалар саны.

Егер қала  $X$  қаласынан және  $Y$  қаласынан жетімді болса, ол ыңғайлылықта *екі рет* есептелетінін ескеріңіз.

Сіздің мақсатыңыз ең үлкен ыңғайлылықты табу.

## Implementation Details

Сізге келесі функцияны асыру қажет:

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

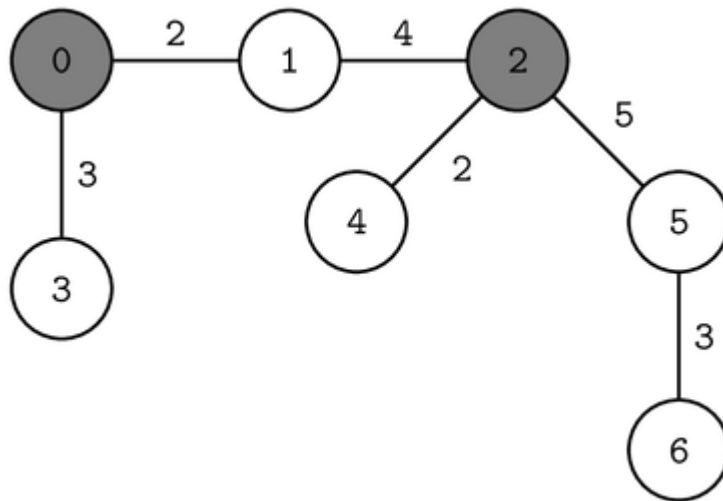
- $N$ : қалалар саны.
- $X, Y$ : екі әр түрлі фестиваль алаңы бар қалалар.
- $K$ : жабылу уақыттардың қосындысының жоғарғы шегі.
- $U, V$ : тікелей жол байланыстарының сипаттайтын, ұзындығы  $N - 1$  жиымдар.
- $W$ : тікелей жолдардың ұзындығын сипаттайтын, ұзындығы  $N - 1$  жиым.
- Бұл функция ең үлкен ыңғайлылықты қайтару қажет.
- Бұл функция әр тестте **бірнеше рет** шақыртылуы мүмкін.

## Example

Келесі шақыртуды қарастырайық:

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

Бұл келесі жол желісіне сәйкес келеді:



Жабылу уақыты келесідей тағайындалсын:

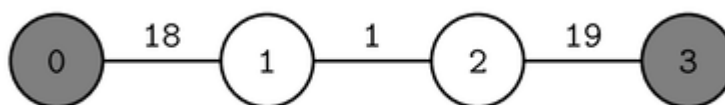
Қала	0	1	2	3	4	5	6
Жабылу уақыты	0	4	0	3	2	0	0

Барлық жабылу уақыттардың қосындысы 9, бұл  $K = 10$ -ды аспайды. 0, 1 және 3 қалалар  $X$  ( $X = 0$ ) қаласынан жетімді, ал 1, 2 және 4 қалалары  $Y$  ( $Y = 2$ ) қаласынан жетімді. Осылайша, ыңғайлылық  $3 + 3 = 6$  болады. Ыңғайлылық 6-дан жоғары бола алмайды, сондықтан функция 6 мәнін қайтаруы керек.

Келесі шақыртуды да қарастырайық:

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

Бұл келесі жол желісіне сәйкес келеді:



Жабылу уақыты келесідей тағайындалсын:

Қала	0	1	2	3
Жабылу уақыты	0	1	19	0

0 қаласы  $X$  ( $X = 0$ ) қаласынан жетімді, ал 2 және 3 қалалары  $Y$  ( $Y = 3$ ) қаласынан жетімді. Осылайша, ыңғайлылық  $1 + 2 = 3$  болады. Ыңғайлылық 3-тен жоғары бола алмайды,

сондықтан функция 3 мәнін қайтаруы керек.

## Constraints

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$  ( $0 \leq j \leq N - 2$  орындалатын әр бір  $j$  үшін)
- $1 \leq W[j] \leq 10^6$  ( $0 \leq j \leq N - 2$  орындалатын әр бір  $j$  үшін)
- Жолдармен кез-келген қаладан кез-келген басқа қалаға жетуге болады.
- $S_N \leq 200\,000$ ,  $S_N$  - әр тесттің `max_score` шақыртулар бойынша  $N$  қосындысы.

## Subtasks

Біз жол желісін **сызықты** деп айтамыз егер  $i$  жолы  $i$  қаласын және  $i + 1$  қаласын байланыстырады (әр бір  $i$  ( $0 \leq i \leq N - 2$  үшін)).

1. (8 points)  $X$  қаласынан  $Y$  қаласына дейінгі жолдың ұзындығы  $2K$ -дан кем емес.
2. (9 points)  $S_N \leq 50$ , жол желісі сызықты.
3. (12 points)  $S_N \leq 500$ , жол желісі сызықты.
4. (14 points)  $S_N \leq 3\,000$ , жол желісі сызықты.
5. (9 points)  $S_N \leq 20$
6. (11 points)  $S_N \leq 100$
7. (10 points)  $S_N \leq 500$
8. (10 points)  $S_N \leq 3\,000$
9. (17 points) Қосымша шектеулер жоқ.

## Sample Grader

$C$ -деп тестер санын айтайық, яғни, `max_score` функцияның шақыртулар саны. Сынақ бағалаушы кірісті келесі форматта оқиды:

- line 1:  $C$

$C$  тестін сипаттамасы келесі.

Сынақ бағалаушы әр тестін сипаттамасын келесі форматта оқиды:

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

Сынақ бағалаушы әр тест үшін бір қатарды келесі форматта шағарды:

- line 1: the return value of `max_score`