

Жабуу убактысы

Венгрия — 0дөн $(N - 1)$ ге чейинки N шаарлары бар өлкө.

Шаарлар $(N - 1)$ эки багытта жол менен байланышкан, 0дөн $(N - 2)$ ге чейин номерленген. Ар бир j үчүн $0 \leq j \leq N - 2$, j жолу $U[j]$ шаарын жана $V[j]$ шаарларын бириктирет жана узундугу $W[j]$, башкача айтканда, ал $W[j]$ убакытта шаарлардын арасында саякаттоого болот. Ар бир жол эки башка шаарды бириктирет жана ар бир жуп шаар эң көп дегенде бир жол менен байланышат.

a жана b эки башка шаардын арасындагы **жол** бул ар башка шаарлардын p_0, p_1, \dots, p_t удаалаштыгы, төмөндөгүдөй:

- $p_0 = a$,
- $p_t = b$,
- ар бир i ($0 \leq i < t$) үчүн p_i жана p_{i+1} шаарларын бириктирген жол бар.

Жолдор аркылуу каалаган шаардан башка шаарга барууга болот, башкача айтканда, ар бир эки шаардын арасында жол бар. Бул жол өзүнчө шаарлардын ар бир жуп үчүн уникалдуу экенин көрсөтүүгө болот.

Жолдун **узундугу** p_0, p_1, \dots, p_t - бул жол боюндагы ырааттуу шаарларды бириктирген t жолдун узундуктарынын суммасы.

Венгрияда көптөгөн адамдар эки чоң шаардагы Негиздөө күнүнө арналган салтанаттарга барышат. Майрамдар аяктагандан кийин алар үйлөрүнө кайтып келишет. Өкмөт чогулган элдин жергиликтүү тургундардын тынчын албоосун каалайт, ошондуктан алар белгилүү бир убакта бардык шаарларды жабууну пландаштырууда. Ар бир шаарга өкмөт тарабынан терс эмес **жабуу убактысы** дайындалат. Өкмөт бардык жабылуу убакыттарынын суммасы K дан ашпоого тийиш деп чечти. Тагыраак айтканда, 0 жана $(N - 1)$ арасындагы ар бир i үчүн, анын ичинде i шаарына дайындалган жабылуу убактысы $c[i]$ терс эмес бүтүн сан болуп саналат. Бардык $c[i]$ суммаларынын суммасы K дан ашпашы керек.

a шаарын жана жабылуу убакыттарынын айрымдарын карап көрөлү. Биз b шаарына a шаарынан **жетүүгө болот** деп айтабыз, эгерде $b = a$, же бул эки шаардын ортосундагы p_0, \dots, p_t жолу (айрыкча $p_0 = a$ жана $p_t = b$) төмөнкү шарттарды канааттандырат:

- жолдун узундугу p_0, p_1 эң көп дегенде $c[p_1]$, жана
- жолдун узундугу p_0, p_1, p_2 эң көп дегенде $c[p_2]$, жана

- ...
- жолдун узундугу $p_0, p_1, p_2, \dots, p_t$ эң көп дегенде $c[p_t]$.

Бул жылы фестивалдын эки негизги жери X шаарында жана Y шаарында жайгашкан. Жабуу убактысынын ар бир дайындоосу үчүн **ыңгайлуулук упай** төмөнкү эки сандын суммасы катары аныкталат:

- X шаарынан жете турган шаарлардын саны.
- Y шаарынан жете турган шаарлардын саны.

Эгер шаарга X шаарынан жетсе жана Y шаарынан жетсе, ал ыңгайлуулук упайына эки эсе эсептелерин эске алыңыз.

Сиздин милдет - жабуу убактысынын айрым дайындоосу менен жетишүүгө мүмкүн болгон максималдуу **ыңгайлуулук упай**ды эсептөө.

Процедуранын чоо-жайы

Сиз төмөнкү процедураны ишке ашырууңуз керек:

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

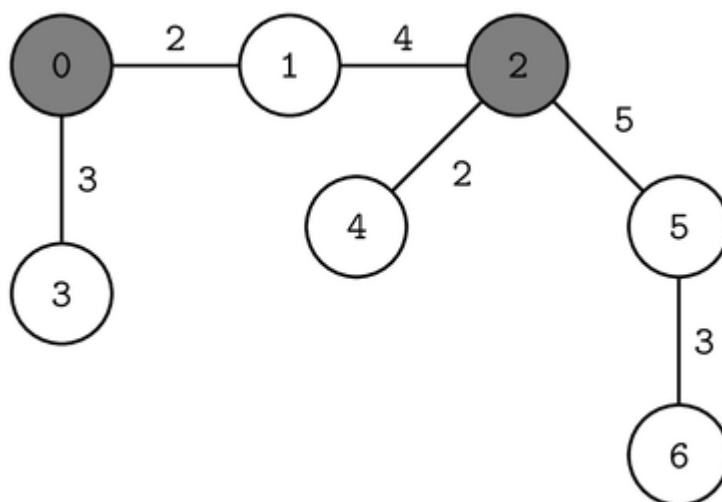
- N : шаарлардын саны.
- X, Y : фестивалдын негизги жерлери бар шаарлар.
- K : жабуу убакыттарынын суммасынын жогорку чеги.
- U, V : жол байланыштарын сүрөттөгөн $(N - 1)$ узундуктагы массивдер.
- W : жолдун узундугун сүрөттөгөн $(N - 1)$ узундуктагы массив.
- Бул процедуранын жабылуу убакыттарынын кээ бир дайындоосу менен жетишүүгө мүмкүн болгон максималдуу ыңгайлуулукту кайтарып бериши керек.
- Бул процедураны ар бир тестте **бир нече жолу** чакырышы мүмкүн.

Мисал

Төмөнкү чакырууну карап көрөлү:

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

Бул төмөнкү жол тармагына туура келет:



Төмөнкүдөй жабылуу убакыттары дайындалды дейли:

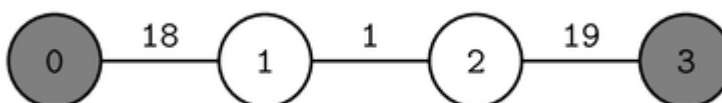
Шаар	0	1	2	3	4	5	6
Жабуу убактысы	0	4	0	3	2	0	0

Баардык жабылуу убакыттарынын суммасы 9 экенине көңүл буруңуз, бул $K = 10$ дон көп эмес. 0, 1 жана 3 шаарларына X шаарынан ($X = 0$), ал эми 1, 2 жана 4 шаарларына Y шаарынан жетүүгө болот ($Y = 2$). Демек, ыңгайлуулук упайы $3 + 3 = 6$. Ыңгайлуулугу 6дан ашкан жабылуу убактысын дайындоо жок, ошондуктан процедура 6 кайтарып бериши керек.

Ошондой эле төмөнкү чакырууну карап көрөлү:

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

Бул төмөнкү жол тармагына туура келет:



Төмөнкүдөй жабылуу убакыттары дайындалды дейли:

Шаар	0	1	2	3
Жабуу убактысы	0	1	19	0

0 шаарына X шаарынан ($X = 0$), ал эми 2 жана 3 шаарларына Y шаарынан ($Y = 3$) жетсе болот. Демек, ыңгайлуулук упайы $1 + 2 = 3$. Ыңгайлуулугу 3төн ашкан жабылуу убактысын

дайындоо жок, ошондуктан процедура 3 кайтарышы керек.

Чектөөлөр

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (ар бир j үчүн $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (ар бир j үчүн $0 \leq j \leq N - 2$)
- Жолдорду колдонуу менен каалаган шаардан башка шаарга барууга болот.
- $S_N \leq 200\,000$, мында S_N - ар бир сыноо учурундагы бардык N чалуулардын `max_score` суммасы.

Кошумча тапшырмалар

i жолу i жана $i + 1$ шаарларын байласа, жол тармагы **сызыктуу** деп айтабыз (ар бир i үчүн $0 \leq i \leq N - 2$).

1. (8 упай) X шаарынан Y шаарына чейинки жолдун узундугу $2K$ дан чоңураак.
2. (9 упай) $S_N \leq 50$, жол тармагы сызыктуу.
3. (12 упай) $S_N \leq 500$, жол тармагы сызыктуу.
4. (14 упай) $S_N \leq 3\,000$, жол тармагы сызыктуу.
5. (9 упай) $S_N \leq 20$
6. (11 упай) $S_N \leq 100$
7. (10 упай) $S_N \leq 500$
8. (10 упай) $S_N \leq 3\,000$
9. (17 упай) Кошумча чектөөлөр жок.

Үлгү Грейдер

C сценарийлердин санын, башкача айтканда, `max_score` чалуулардын санын белгилесин.

Үлгү грейдер киргизүүнү төмөнкү форматта окуйт:

- 1-сап: C

C сценарийлеринин сүрөттөмөлөрү төмөнкүдөй.

Үлгү грейдер ар бир сценарийдин сүрөттөмөсүн төмөнкү форматта окуйт:

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

Үлгү грейдер ар бир сценарий үчүн төмөнкү форматта бир сапты басып чыгарат:

- 1 - сап : максималдуу_упайдын кайтаруу мааниси