Vrijeme zatvaranja

Mađarska je država s N gradova, označenih brojevima od 0 do $N-1$.

Gradovi su povezani s $N-1$ *dvosmjernih* cesta, označenih brojevima od 0 do $N-2$. Za svaki j takav da $0 \leq j \leq N-2$, cesta j povezuje grad $U[j]$ i grad $V[j]$ te ima duljinu $W[j]$, točnije, dopušta putovanje između tih dvaju gradova u $W[j]$ jedinica vremena. Svaka cesta povezuje dva različita grada, te svaki par gradova povezan je najviše jednom cestom.

Put između dva različita grada a i b niz je $p_0, p_1, \ldots, p_t$ različitih gradova, takav da:

- $p_0 = a$,
- $p_t = b$,
- za svaki i ($0 \leq i < t$), postoji cesta koja povezuje gradove p_i te p_{i+1}.

Moguće je putovati iz bilo kojeg grada u bilo koji drugi grad koristeći dane ceste, točnije, postoji put između svaka dva različita grada. Može se pokazati da je taj put jedinstven za svaki par različitih gradova.

Duljina puta $p_0, p_1, \ldots, p_t$ zbroj je duljina svih t cesta koje povezujuj susjedne gradove duž puta.

U Mađarskoj, ljudi često putuju kako bi sudjelovali u Danu osnutka u jednom od dva velika grada. Jednom kada se proslava završi, ljudi se moraju vratiti svojim domovima. Vlada, kako bi spriječila preveliku buku koja bi ometala domaće stanovništvo, odlučila je zatvoriti neka gradove u nekim trenutcima. Vlada je odlučila da zbroj vremena zatvaranja svih gradove ne smije biti veći od K. Točnije, za svaki i između 0 i $N-1$, uključivo, vrijeme zatvranja pridruženo gradu i može se prikazati kao nenegativan broj $c[i]$. Zbroj svih $c[i]$ ne smije biti veća od K.

Promatrajmo grad a i neko dodjeljivanje vremena zatvaranja. Grad b je **dostupan** od grada a ako i samo ako je ili $b = a$, ili put $p_0, \ldots, p_t$ između tih dvaju gradova (u tom slučaju vrijedi $p_0 = a$ te $p_t = b$) zadovoljava sljedeće uvjete:

- duljina puta p_0, p_1 je najviše $c[p_1]$, i
- duljina puta p_0, p_1, p_2 je najviše $c[p_2]$, i
- $\ldots$
- duljina puta $p_0, p_1, p_2, \ldots, p_t$ je najviše $c[p_t]$.

Ove godine, dva glavna festivala održavat će se u gradovima X te Y. Za svako dodjeljivanje vremena zatvaranja, **prikladnost** definiramo kao zbroj sljedećih dvaju brojeva:

- Broj gradova dostupnih od grada X.
- Broj gradova dostupnih od grada Y.

Primijetite da ako je grad dostupan od grada X i od dostupan od grada Y, tada se broji *dvaput* u prikladnosti tog dodjeljivanja.

Vaš je zadatak izračunati najveću moguću prikladnost nekog dodjeljivanja vremena zatvaranja unutar danih uvjeta.

Implementacijski detalji

Morate implementirati sljedeću funkciju

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

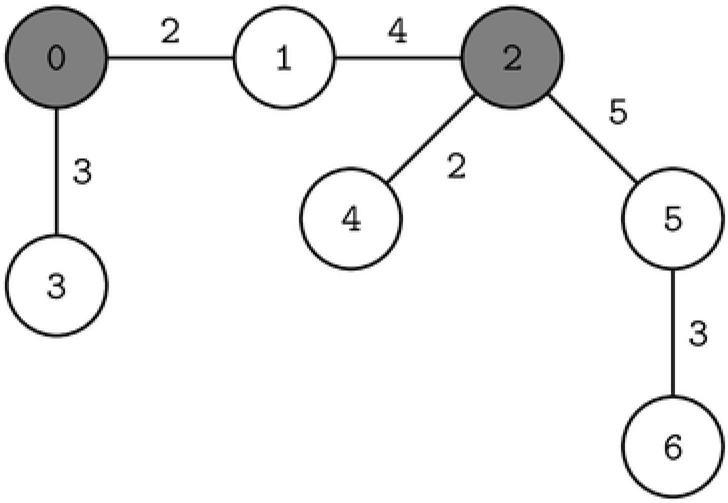
- N: broj gradova.
- X, Y: oznake dva velika grada
- K: gornja granica na zbroj vremena zatvaranja.
- U, V: nizovi duljine $N-1$ koji opisuju gradove koji povezuju ceste.
- W: niz duljine $N-1$ koji opisuje duljine cesta.
- Ova funkcija mora vratiti najveću moguću prikladnost koja se postiže nekim dodjeljivanjem vremena zatvaranja pod danim uvjetima.
- Ova funkcija može biti pozvana **više puta** tijekom istog primjera.

Primjer

Promatrajte sljedeći poziv funkcije.

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

Pripadajuća cestovna mreža izgleda ovako:



Pretpostavimo da su gradovima pridružena sljedeća vremena zatvaranja.

Grad	0	1	2	3	4	5	6
Vrijeme zatvaranja	0	4	0	3	2	0	0

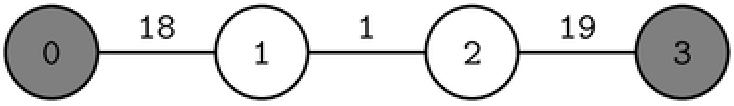
Primijetite da je zbroj svih vremena zatvaranja 9, što nije veće od $K = 10$.

Gradovi 0, 1, i 3 dostupni su od grada X ($X = 0$), dok gradovi 1, 2, i 4 dostupni su od grada Y ($Y = 2$). Stoga, prikladnost iznosi $3 + 3 = 6$. Ne postoji dodjeljivanje vremena zatvaranja koje postiže prikladnost veću do 6, stoga funkcija mora vratiti 6.

Također, promatrajte sljedeći poziv:

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

Pripadajuća cestovna mreža izgleda ovako:



Pretpostavimo da su gradovima pridružena sljedeća vremena zatvaranja.

City	0	1	2	3
Closing time	0	1	19	0

Grad 0 dostupan je od grada X ($X = 0$), dok su gradovi 2 i 3 dostupni od grada Y ($Y = 3$). Stoga, prikladnost iznosi $1 + 3 = 6$. Ne postoji dodjeljivanje vremena zatvaranja koje postiže prikladnost veću do 3, stoga funkcija mora vratiti 3.

Ograničenja

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (za svaki j takav da je $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (za svaki j takav da je $0 \leq j \leq N - 2$)
- Moguće je putovati iz bilo kojeg grada u bilo koji koristeći ceste.
- $S_N \leq 200\,000$, gdje S_N označava sumu svih N po svim pozivima funkcije `max_score` u primjeru.

Podzadaci

Neka cestovna mreža je **linearna** ako i-ta cesta povezuje gradove i te $i + 1$ (za svaki i takav da je $0 \leq i \leq N - 2$).

1. (8 bodova) Duljina puta između gradova X i Y veća je od $2K$.
2. (9 bodova) $S_N \leq 50$, cestovna mreža je linearna.
3. (12 bodova) $S_N \leq 500$, cestovna mreža je linearna.
4. (14 bodova) $S_N \leq 3\,000$, cestovna mreža je linearna.
5. (9 bodova) $S_N \leq 20$
6. (11 bodova) $S_N \leq 100$
7. (10 bodova) $S_N \leq 500$
8. (10 bodova) $S_N \leq 3\,000$
9. (17 bodova) Nema dodatnih ograničenja.

Probni ocjenjivač

Neka C označava broj slučajeva, točnije, broj pozivanja funkcije `max_score`. Probni ocjenjivač učitava ulaz u sljedećem obliku:

- 1. redak : C

Slijede opisi C slučajeva.

Probni ocjenjivač učitava opise svakog slučaja u sljedećem obliku.

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

Probni ocjenjivač ispisuje jedan redak za svaki slučaj, u sljedećem obliku:

- 1. redak : vrijednost koju je vratila funkcija `max_score`