



## Zatvaranje

Mađarska je zemlja sa  $N$  gradova, numerisanih od 0 do  $N - 1$ .

Gradovi su spojeni sa  $N - 1$  *dvosmjernih* ulica, numerisanih od 0 do  $N - 2$ . Za svako  $j$  takvo da  $0 \leq j \leq N - 2$ , cesta  $j$  spaja grad  $U[j]$  i grad  $V[j]$  i ima dužinu  $W[j]$ , odnosno, daje osobi mogućnost da putuje između dva grada u  $W[j]$  jedinica vremena. Svaka cesta spaja dva grada i svaki par gradova je konektovan sa najviše jednom cestom.

**Put** između dva različita grada  $a$  i  $b$  je sekvenca  $p_0, p_1, \dots, p_t$  različitih gradova, takva da:

- $p_0 = a$ ,
- $p_t = b$ ,
- za svako  $i$  ( $0 \leq i < t$ ), ima cesta koja spaja gradove  $p_i$  i  $p_{i+1}$ .

Moguće je putovati iz bilo kojeg grada u bilo koji grad koristeći ceste, tačnije, postoji put između bilo koja dva različita grada. Primjetite da je ovaj put unikatan za svaki par različitih gradova.

**Dužina** puta  $p_0, p_1, \dots, p_t$  je ukupna suma dužina  $t$  cesta koja spaja uzastopne gradove u putu.

U Mađarskoj, mnogo ljudi putuje da bi prisustvovali Danu Osnivanja u dva velika grada. Nakon priređenih zabava se vraćaju kući. Vlada želi da ne ometa lokalno stanovništvo, te planiraju da zaključaju sve gradove u nekim trenucima. Svakom gradu će biti dodjeljen ne-negativno **vrijeme zatvaranja** od vlade. Vlada je odlučila da suma svih vremena zatvaranja ne smije biti veća od  $K$ . Preciznije, za svako  $i$  između 0 i  $N - 1$  (inkluzivno), vrijeme zatvaranja dodjeljeno gradu  $i$  je ne-negativan cijeli broj  $c[i]$ . Suma svih  $c[i]$  ne smije biti veća od  $K$ .

Uzmimo za primjer grad  $a$  i određena vremena zatvaranja za taj grad. Možemo reći da se u grad  $b$  **može doći** iz grada  $a$  ako i samo ako je  $b = a$ , ili put  $p_0, \dots, p_t$  između ova dva grada (odnosno  $p_0 = a$  i  $p_t = b$ ) zadovoljava sljedeće uslove:

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

Ove godine dva glavna grada festivala se odvijaju u gradovima  $X$  i u gradu  $Y$ . Za svako dodjeljeno vrijeme zatvaranja, **bodovi ugodnosti** su definisani kao suma sljedeća dva broja:

- Broj gradova dostupnih iz grada  $X$ .

- Broj gradova dostupnih iz grada  $Y$ .

Primjetite da ako je neki grad dostupan iz grada  $X$  i dostupan je iz grada  $Y$ , on se broji *dva puta* u *bodovima ugodnosti*.

Vaš zadatak je da izračunate najveći mogući broj *bodova ugodnosti* za neka vremena zatvaranja.

## Detalji implementacije

Trebate implementirati sljedeću funkciju.

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

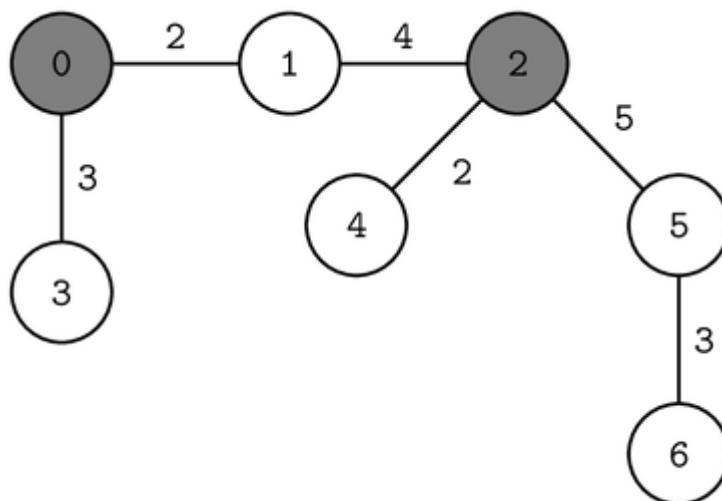
- $N$ : broj gradova.
- $X, Y$ : gradovi sa glavnim festivalima.
- $K$ : najveća moguća suma svih vremena zatvaranja.
- $U, V$ : nizovi dužine  $N - 1$  koji opisuju povezanost cesta.
- $W$ : niz dužine  $N - 1$  koji opisuje dužine cesta.
- Ova funkcija treba da vrati najveći mogući broj *bodova ugodnosti* za neka vremena zatvaranja.
- Ova se funkcija može zvati **više puta** u svakom testnom slučaju.

## Primjer

Uzmimo za primjer sljedeći poziv ove 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])
```

Ovo odgovara sljedećoj mreži cesta:



Pretpostavimo da su vremena zatvaranja sljedeća:

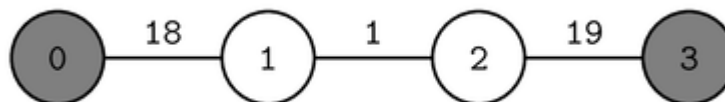
Grad	0	1	2	3	4	5	6
Vrijeme zatvaranja	0	4	0	3	2	0	0

Primijetite da je suma svih vremena zatvaranja 9, što nije više od  $K = 10$ . Gradovi 0, 1, i 3 su dostupni iz grada  $X$  ( $X = 0$ ), dok gradovi 1, 2, i 4 su dostupni iz grada  $Y$  ( $Y = 2$ ). Dakle, bodovi ugodnosti su  $3 + 3 = 6$ . Ne postoji postava vremena zatvaranja takva da su bodovi ugodnosti veći od 6, tako da ova funkcija treba vratiti 6.

Također razmotrimo sljedeći poziv:

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

Što odgovara sljedećoj mreži cesta:



Pretpostavimo da su vremena zatvaranja sljedeća:

City	0	1	2	3
Closing time	0	1	19	0

Grad 0 je dostupan iz grada  $X$  ( $X = 0$ ), dok su gradovi 2 i 3 dostupni iz grada  $Y$  ( $Y = 3$ ). Dakle bodovi ugodnosti su  $1 + 2 = 3$ . Ne postoji postava vremena zatvaranja takva da su bodovi ugodnosti veći od 3, tako da bi ova funkcija trebala vratiti rezultat 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 svako  $j$  takvo da  $0 \leq j \leq N - 2$ )
- $1 \leq W[j] \leq 10^6$  (za svako  $j$  takvo da  $0 \leq j \leq N - 2$ )
- Moguće je putovati iz bilo kojeg grada u bilo koji grad koristeći ceste.
- $S_N \leq 200\,000$ , gdje  $S_N$  je suma svih  $N$  tokom svih poziva funkcije `max_score` u jednom testnom primjeru.

## Podzadaci

Mrežu cesta nazivamo **linearom** ako cesta  $i$  spaja gradove  $i$  i  $i + 1$  (za svako  $i$  tako da  $0 \leq i \leq N - 2$ ).

1. (8 bodova) Dužina puta od  $X$  do  $Y$  je veća od  $2K$ .
2. (9 bodova)  $S_N \leq 50$ , mreža cesta je linearna.
3. (12 bodova)  $S_N \leq 500$ , mreža cesta je linearna.
4. (14 bodova)  $S_N \leq 3\,000$ , mreža cesta 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) Bez dodatnih ograničenja.

## Primjer gradera

Neka neko  $C$  predstavlja broj scenarija, tačnije, broj poziva na `max_score`. Primjer gradera čita ulaz u sljedećem formatu:

- linija 1:  $C$

Slijede opisi scenarija  $C$ .

Primjer gradera čita opise svakog scenarija u sljedećem formatu:

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

Primjer gradera printa jednu liniju za svaki scenario u sljedećem formatu:

- linija 1: vraća vrijednost `max_score`