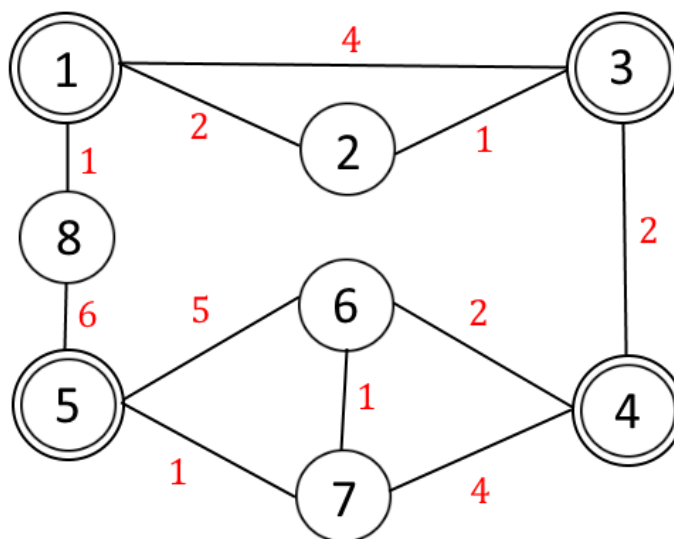


Mreža stanica (mreža)

Javni prijevoz je krvotok jednoga grada. U malom gradiću X prije nije postojao javni prijevoz, ali je došlo vrijeme da se uvede kako bi omogućio bolji transport građana. Popularni prijevoznik Y je došao u grad X s takvom propozicijom. Njihov cilj je da uvežu grad tako da se transport vrši što efikasnije.

Kao novi zaposlenik kompanije Y, na vama je zadatak da napravite algoritam koji pronađe minimalnu ukupnu dužinu uvezane mreže grada X. Preciznije, imate specifične dijelove grada koji su unaprijed određeni kao stanice i oni se trebaju uvezati. Ono što se ne zna je kako najbolje povezati dvije stanice i kako najbolje povezati cjelokupnu mrežu stanica. Kod povezivanja dvije stanice kraća putanja se više cijeni tj. uvijek najkraći put između dvije stanice pravimo bez posjećivanja drugih stanica, dok je kod povezivanja mreže stanica bitno da se sve stanice povežu tako da ukupna dužina bude minimalna. Dijelovi grada koji nisu stanice mogu ostati i nepovezani javnim prijevozom.

Vama su dostupni podaci u vidu grafa koji pokazuje sve moguće putanje prijevoza. Graf sadrži tačke, putanje i dužine putanja. Tačke mogu biti obične tačke, kroz kojih prijevoz može prolaziti, ali i stanice, na kojima prijevoz mora stati. Obične tačke su označene s jednim, a stanice s dva kruga. Broj koji se nalazi u krugu predstavlja ID stanice. Linija između dvije tačke predstavlja jednu putanju i ima određenu dužinu. Što je manja dužina, to bolje.



Bitno je naglasiti da veza dvije stanice sadrži sopstvene obične tačke koje se ne mogu koristiti da vežu bilo koju drugu stanicu. Na datom primjeru tačke 6 i 7 mogu vezati samo stanice 5 i 4, kao i sve tačke između 5 i 4, a nikada tačku koja se veže za neku drugu stanicu. Između dvije stanice se mogu nalaziti mnoge obične tačke koje se povezuju među sobom. Poenta je da se mreža koja se nalazi između dvije stanice gleda kao izolovana mreža od ostalih mreža.

Svi putevi su dvosmjerni. Finalni prijedlog transporta (tj. puteva koji će se napraviti) mora zadovoljavati uslov da je sa svake stanice moguće doći do bilo koje druge.

Zadatak

Vaš zadatak je da napravite program koji implementira tri funkcije:

```
void Stanice(int N, int S[], int Q);
```

Gdje N označava broj stanica, Q predstavlja broj veza između dvije stanice, dok je S niz od tačno N elemenata gdje $S[i]$ predstavlja indeks i -te stanice u gradu. Nakon toga, tačno Q puta će se pozvati sljedeća funkcija koju također vi implementirate:

```
void PoveziStanice(int S1, int S2, int T, int V, int L[], int A[],  
int B[], int C[]);
```

Gdje $S1$ i $S2$ označavaju stanice koje se povezuju nekom izolovanom podmrežom, a ostali parametri opisuju tu podmrežu. T predstavlja broj veznih tačaka između te dvije stanice, a V predstavlja broj veza, tj. puteva između te dvije stanice. L je niz od tačno T članova koji predstavljaju indekse veznih tačaka koje se nalaze u ovoj izolovanoj mreži, dok su nizovi A , B i C od svaki po tačno V članova opis veza u toj mreži. $A[i]$ i $B[i]$ označavaju krajnje tačke i -te veze dok $C[i]$ označava dužinu te veze/puta. Nakon svih Q poziva, poziva se i zadnja funkcija:

```
int UkupnaDuzina();
```

Ova funkcija ne prima nikakve parametre jer je opis grafa poslan preko prethodnih funkcija, ali zato vraća ukupnu dužinu mreže koja bi bila optimalna kada se u obzir uzmu svi zahtjevi iz teksta.

Primjeri

Obzirom na količinu parametara koje funkcije primaju, kao i broj njihovih poziva, javne primjere je puno lakše objasniti uz prikaz ulaznih i izlaznih datoteka koje koriste *grader*-i na serveru nego simulacijom poziva funkcija. Kod samih *gradera*-a će biti dostupan u sklopu postavke zadatka na serveru za ocjenjivanje. Kako biste lakše razumjeli šta predstavljaju ulazni podaci, u nastavku testka je dodan i opis ulaznih i izlaznih datoteka.

Primjer br. 1

Ulaz:	Izlaz:
4 1 3 5 4 4 1 3 1 3 2 1 2 2	9

1 3 4	
2 3 1	
3 4 0 1	
3 4 2	
5 4 2 5	
6 7	
5 6 5	
6 4 2	
6 7 1	
5 7 1	
4 7 4	
1 5 1 2	
8	
1 8 1	
8 5 6	

Objašnjenje: Dati primjer predstavlja sliku iznad. Može se zaključiti da je najkraći put između stanica 1 i 3 dužine 3, između 3 i 4 dužine 2, između 4 i 5 dužine 4, između 5 i 1 dužine 7. Da bi se od te stanice načinila uvezana mreža potrebno je spojiti stanice 1 i 3, 3 i 4, 4 i 5. Stanice 5 i 1 se ne povezuju. Ukupna dužina mreže je $3 + 2 + 4$, odnosno 9.

Primjer br. 2

<i>Ulaz:</i>	<i>Izlaz:</i>
4	12
1 4 7 9	
3	
1 4 2 5	
2 3	
1 2 5	
1 3 2	
2 3 1	
2 4 2	
3 4 5	
4 7 2 4	
5 6	
4 5 2	
5 7 1	
4 6 1	
6 7 3	
4 9 1 2	
8	
4 8 2	
8 9 2	

Objašnjenje: Da bi se mreža uvezala potrebno je povezati stanice 1 i 4 (dužine 5), stanice 4 i 9 (dužine 4) i stanice 4 i 7 (dužine 3), što ukupno daje 11.

Format ulaza i izlaza

Na prvoj liniji ulaza se nalazi broj N , koji označava broj stanica. Sljedećih N brojeva predstavljaju brojeve stanica. Na sljedećoj liniji se nalazi broj Q koji predstavlja broj veza između dvije stanice. U sljedećih Q dijelova se nalaze sljedeći podaci:

- Prva linija sadrži četiri broja; prva dva broja označavaju stanice koje se vežu, treći broj predstavlja broj T , tj. broj veznih tačaka između dvije stanice, dok četvrti broj predstavlja broj V , tj. broj veza između te dvije stanice;
- Druga linija sadrži T brojeva koji predstavljaju brojeve veznih tačaka;
- Sljedećih V linija sadrže veze, tj. prva dva broja označavaju dvije tačke dok treći broj označava dužinu.

Na jednoj liniji izlaza ispisan je jedan cijeli broj koji označava ukupnu dužinu povezane mreže.

Bodovanje i ograničenja

Zadatak će biti testiran na deset testnih primjera od kojih svaki nosi deset bodova.

Za sve testne primjere vrijede sljedeća ograničenja: $N \leq 200$, $Q \leq 400$, $T \leq 100$, $V \leq 250$. Ukupan broj tačaka na grafu može dostizati 30000.

Vremenska i memorijska ograničenja su data u sistemu za ocjenjivanje.