

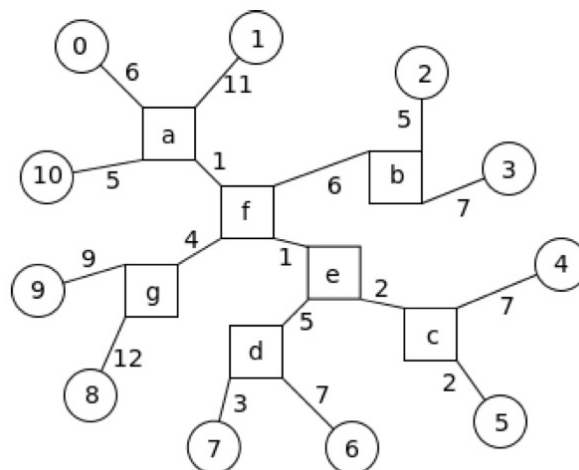
Towns

U Kazahstanu postoji N sela, koja su numerisana od 0 do $N - 1$. Takođe postoji i nepoznat broj gradova. Sela i gradovi Kazahstana se zajedničkim imenom nazivaju *naselja*.

Sva naselja Kazahstana su povezana jednom mrežom dvosmjernih puteva. Svaki put povezuje dva različita naselja, i svaki par naselja je direktno povezan najviše jednim putem. Za svaki par naselja a i b postoji jedinstven način da se, koristeći navedne puteve, dođe od naselja a do naselja b , sve dok se niti jedan od puteva ne koristi više od jednom.

Poznato je da je svako selo direktno povezano sa tačno jednim drugim naseljem, a svaki grad je direktno povezan sa tri ili više naselja.

Sljedeća slika prikazuje mrežu od **11** sela i **7** gradova. Sela su prikazana kao krugovi i označena su cijelim brojevima, a gradovi su prikazani kao kvadrati i označeni su slovima.



Dužina svakog puta je pozitivan cio broj. Udaljenost između dva naselja je minimalna suma dužina puteva koji se moraju preći kako bi se došlo od jednog naselja do drugog.

Za svaki grad C označićemo sa $r(C)$ udaljenost do sela koje se nalazi na najvećoj udaljenosti od tog grada. Grad C nazivamo centrom (hub) ako je udaljenost $r(C)$ najmanja među svim gradovima.

Udaljenost između centra i sela koje je najudaljenije od centra označimo sa R . Dakle, R predstavlja najmanju vrijednost $r(C)$ među svim gradovima.

U gornjem primjeru najudaljenije selo od grada a je selo 8 i udaljenost između njih je $r(a) = 1 + 4 + 12 = 17$. Za grad g je takođe $r(g) = 17$. (Jedno od sela koje je najudaljenije za grad g je selo 6 .) Jedini centar u gornjem primjeru je grad f , sa $r(f) = 16$. Dakle, u gornjem primjeru R je 16 .

Izbacivanjem centra mreža se dijeli na više povezanih komponenti. Centar je *balansiran* ako svaka od

tih komponenti sadrži najviše $\lfloor N/2 \rfloor$ sela. (Dakle, ne brojimo gradove.) Sa $\lfloor x \rfloor$ označen je najveći cio broj ne veći od x .

U gornjem primjeru grad f je centar. Ako izbacimo grad f , mreža se dijeli na četiri povezane komponente. Te četiri komponente odgovaraju sljedećim skupovima sela: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$ i $\{8, 9\}$. Nijedna od tih komponenti nema više od $\lfloor 11/2 \rfloor = 5$ sela, dakle grad f je balansirani centar.

Zadatak

Na početku, jedina informacija koju imate o mreži naselja i puteva je broj sela N . Ne znate broj gradova. Takođe, ne znate ništa o rasporedu puteva. Dodatne informacije možete dobiti jedino postavljanjem upita o udaljenostima između parova sela.

Vaš je zadatak da odredite:

- U svim podzadacima: udaljenost R .
- U podzadacima 3 do 6: da li u mreži postoji balansirani centar.

Potrebno je implementirati funkciju `hubDistance`. Ocjenjivač će pri jednom izvršavanju vašeg programa evaluirati više test primjera. Broj test primjera u jednom izvršavanju je najviše **100**. Za svaki test primjer ocjenjivač će pozvati funkciju `hubDistance` tačno jednom. Obratite pažnju na to da vaša funkcija treba da inicijalizuje sve potrebne promenljive pri svakom pozivu.

- `hubDistance(N, sub)`
 - N : ukupan broj sela.
 - `sub`: redni broj podzadatka (objašnjeno u sekciji Podzadaci).
 - Ako je `sub` 1 ili 2 - funkcija može vratiti bilo R ili $-R$.
 - Ako je `sub` veći od 2 - ako postoji balansirani centar funkcija mora vratiti R , inače vraća $-R$.

Vaša funkcija `hubDistance` može dobiti informacije o mreži puteva pozivima funkcije `getDistance(i, j)`. Ova funkcija vraća udaljenost između sela i i j . Obratite pažnju na to da za $i = j$ funkcija vraća **0**. Za neispravne argumente takođe vraća **0**.

Podzadaci

Za svaki test primjer:

- N je između **6** i **110**, uključivo.
- Udaljenost svaka dva različita sela je između **1** i **1 000 000**, uključivo.

Broj upita koje vaš program smije postaviti je ograničen. Ovo ograničenje je različito u različitim podzadacima, kao što je prikazano u tabeli. Čim vaš program prekorači ograničenje za broj upita, njegovo izvršavanje će biti prekinuto i smatraće se da je dao netačno rješenje.

podzadatak	bodovi	broj upita	naći balansirani centar	dodatna ograničenja
1	13	$\frac{N(N-1)}{2}$	NE	nema
2	12	$\lceil \frac{7N}{2} \rceil$	NE	nema
3	13	$\frac{N(N-1)}{2}$	DA	nema
4	10	$\lceil \frac{7N}{2} \rceil$	DA	svaki grad je povezan s <i>tačno</i> tri puta
5	13	$5N$	DA	nema
6	39	$\lceil \frac{7N}{2} \rceil$	DA	nema

Sa $\lceil x \rceil$ označen je najmanji cio broj ne manji od x .

Ocjenjivač (grader)

Primjetite da je broj podzadatka dio ulaznih podataka. Ocjenjivač se ponaša drugačije za različite podzadatke.

Ocjenjivač čita ulazne podatke iz fajla `towns.in` u sljedećem formatu:

- red 1: Broj podzadatka i ukupan broj test primjera.
- red 2: N_1 , broj sela u prvom test primjeru.
- narednih N_1 redova: j -ti broj ($1 \leq j \leq N_1$) u i -toj od ovih redova ($1 \leq i \leq N_1$) je udaljenost između sela $i - 1$ i $j - 1$.
- Slijede ostali test primjeri, zadati u istom formatu kao i prvi test primjer.

Za svaki test primjer, ocjenjivač štampa povratnu vrijednost funkcije `hubDistance` i broj poziva te funkcije u posebnim linijama.

Ulazni fajl koja odgovara gornjem primjeru je:

```

1 1
11
0 17 18 20 17 12 20 16 23 20 11
17 0 23 25 22 17 25 21 28 25 16
18 23 0 12 21 16 24 20 27 24 17
20 25 12 0 23 18 26 22 29 26 19
17 22 21 23 0 9 21 17 26 23 16
12 17 16 18 9 0 16 12 21 18 11
20 25 24 26 21 16 0 10 29 26 19
16 21 20 22 17 12 10 0 25 22 15
23 28 27 29 26 21 29 25 0 21 22
20 25 24 26 23 18 26 22 21 0 19
11 16 17 19 16 11 19 15 22 19 0

```

Ovaj format je različit od navođenja liste puteva. Primjetite da smijete mijenjati ocjenjivač tako da koristite drugačije formate ulaza.