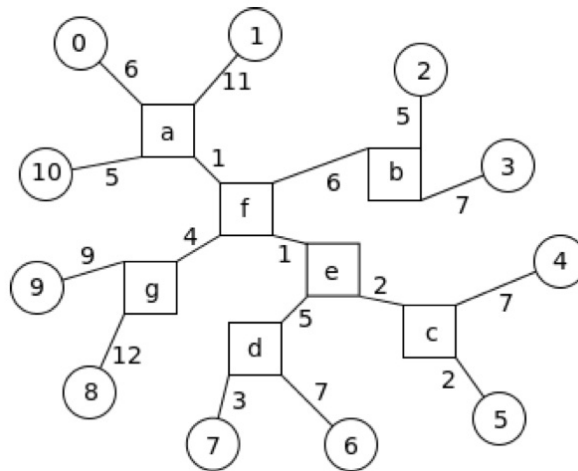


Városok

Kazahsztánban N kisváros van, 0 -tól $N - 1$ -ig sorszámozva, ismeretlen számú nagyváros van. Mindkettőt településnek hívjuk.

A településeket kétirányú utak kötik össze. Bármely két település között legfeljebb egy közvetlen út van. Minden kisvárosból egyetlen út vezet egy másik településhez. Minden nagyvárosból legalább három másik településhez vezet közvetlen út. Továbbá bármely két település között pontosan egy útvonal van.

A következő ábra egy **11** kisvárost és **7** nagyvárost tartalmazó hálózatot mutat. A kisvárosokat körök, a nagyvárosokat négyzetek jelölik.



Minden közvetlen út hossza egy egész szám. Két település távolsága a közöttük vezető útvonal közvetlen útjai hosszának összege.

Minden C nagyvárosra $r(C)$ jelöli a tőle legmesszebb levő kisváros távolságát. A C várost *hub*-nak nevezzük, ha $r(C)$ értéke a legkisebb, jelölje R a legkisebb $r(C)$ értéket!

A fenti példában az a nagyvárostól legmesszebb a **8** kisváros van, a távolságuk $r(a) = 1 + 4 + 12 = 17$. A g nagyvárosra $r(g) = 17$. (A g nagyvárostól legmesszebb levő egyik kisváros a 6.) Egyetlen hub van, az f , amelyre $r(f) = 16$. Tehát R egyenlő **16**.

A hubot eltávolítva, a hálózat több összefüggő részre bomlik. Egy hubot kiegyensúlyozónak nevezzük, ha mindegyik rész legfeljebb $\lfloor N/2 \rfloor$ kisvárost tartalmaz. (Megjegyezzük, hogy a részekben nem számoljuk a nagyvárosokat.) Az $\lfloor x \rfloor$ az x alsó egészrészét jelenti.

A példában f hub. Ha f -et eltávolítjuk, akkor négy összefüggő rész keletkezik, amelyek a következő kisvárosokat tartalmazzák: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$ és $\{8, 9\}$. Egyikük elemszáma sem több, mint $\lfloor 11/2 \rfloor = 5$, tehát f kiegyensúlyozó hub.

Feladat

Kezdetben csak a kisvárosok N számát ismered. Minden további információt kérdésekkel szerezhetsz,

amelyekben két kisváros távolságára kérdezhetsz rá. Nem ismered sem a nagyvárosok számát, sem a hálózat szerkezetét.

Határozd meg:

- Minden részfeladatban: az R értékét.
- A 3 - 6 részfeladatban: van-e kiegyensúlyozó hub.

A `hubDistance` függvényt kell megvalósítanod. Az értékelő egy futás alatt több tesztet futtat, legfeljebb **100**-at. Minden tesztet pontosan egyszer hívja meg a `hubDistance` függvényt. Figyelj arra, hogy minden hívás előtt megfelelően inicializáld a változóidat!

- `hubDistance(N, sub)`
 - N : a kisvárosok száma.
 - `sub`: részfeladat sorszám (kifejtve a Részfeladatok részben).
 - Ha `sub` 1 vagy 2, a függvény R vagy $-R$ értékét is adhatja.
 - Ha `sub` nagyobb 2-nél, akkor ha van kiegyensúlyozó hub, akkor R , egyébként $-R$ értékét adja!

A `hubDistance` függvény `getDistance(i, j)` hívásokkal szerezhet információt, ami az i . és a j . kisváros távolságát adja meg. Ha i és j egyenlő, akkor **0**-t ad. Ha érvénytelen az argumentum, akkor is **0**-t ad.

Részfeladatok

Minden tesztet:

- N legalább **6** és legfeljebb **110**.
- Két kisváros távolsága 1 és 1,000,000 közötti.

A kérdések száma korlátozott, ami részfeladatonként változik az alábbi táblázat szerint. Ha ezt túlléped, akkor az értékelő megszakítja a futást és `incorrect answer` választ ad.

részfeladat	pont	kérdések száma	van-e kiegyensúlyozó hub	további korlátok
1	13	$\frac{n(n-1)}{2}$	NEM	nincs
2	12	$\lceil \frac{7n}{2} \rceil$	NEM	nincs
3	13	$\frac{n(n-1)}{2}$	IGEN	nincs
4	10	$\lceil \frac{7n}{2} \rceil$	IGEN	minden nagyvárosból <i>pontosan</i> három közvetlen útszakasz vezet ki
5	13	$5n$	IGEN	nincs
6	39	$\lceil \frac{7n}{2} \rceil$	IGEN	nincs

Az $\lceil x \rceil$ az x felső egészrészze.

Megvalósítás

A `towns.c`, `towns.cpp`, `towns.pas` vagy `towns.java` file-t kell beadnod, amely tartalmazza a megoldásod!

C/C++ program (include `towns.h` legyen a file első sorában)

```
int hubDistance(int N, int sub);
```

Pascal programs (implement the described method in unit `towns`)

```
function hubDistance(N, sub : longint) : longint;
```

Java programs (implement the described method in public class `towns`)

```
public int hubDistance(int N, int sub);
```

Minta értékelő

A részfeladat sorszám része az inputnak. A minta értékelő a `towns.in` file-ból olvas:

- Az 1. sor: A részfeladat sorszáma és a tesztesetek száma.
- A 2. sor: N_1 , a kisvárosok száma az első tesztesetben.
- A következő N_1 sor: A j . szám ($1 \leq j \leq N_1$) az i . sorban az ($1 \leq i \leq N_1$) az $i - 1$. és a $j - 1$. kisváros távolsága.
- Ezt követi a többi teszteset hasonló formában.

Az értékelő minden tesztesetre kiírja a `hubDistance` értékét és a kérdések számát, külön sorba.

A példának megfelelő input file:

```
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
```