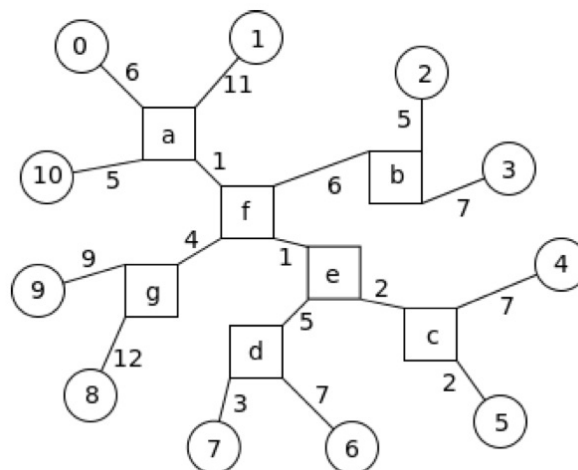


Osady (Towns)

W Kazachstanie jest N osad ponumerowanych od 0 do $N - 1$. W tym kraju jest też pewna, nieznaną liczbą miast. Osady i miasta nazywamy wspólnie *miejscościami*.

Wszystkie miejscowości w Kazachstanie są połączone jedną siecią dwukierunkowych dróg. Każda droga łączy dwie różne miejscowości. Każda para miejscowości jest bezpośrednio połączona co najwyżej jedną drogą. Wiadomo, że każda osada jest połączona bezpośrednio z dokładnie jedną inną miejscowością, a każde miasto jest połączone bezpośrednio z co najmniej trzema miejscowościami. O sieci dróg wiemy ponadto, że dla każdej pary miejscowości a i b istnieje dokładnie jeden sposób przejazdu pomiędzy nimi, jeśli tylko każda z dróg jest użyta co najwyżej raz.

Rysunek poniżej przedstawia sieć złożoną z 11 osad i 7 miast. Osady są oznaczone kółkami z całkowitoliczbowymi etykietami, natomiast miasta są oznaczone kwadratami z etykietami, które są literami.



Każda droga ma dodatnią, całkowitoliczbową długość. Odległość pomiędzy dwiema miejscowościami jest równa minimalnej sumie długości dróg do pokonania tak, żeby się dostać z jednej do drugiej.

Dla każdego miasta C , niech $r(C)$ będzie odległością do najbardziej odległej od C osady. Miasto C nazywamy *centrum*, jeśli odległość $r(C)$ jest najmniejsza spośród wszystkich miast. Odległość między centrum i najbardziej odległą osadą oznaczamy przez R . Tak więc R jest najmniejszą wartością spośród wszystkich $r(C)$.

W powyższym przykładzie najbardziej odległą osadą od miasta a jest osada 8 , a odległość między nimi wynosi $r(a) = 1 + 4 + 12 = 17$.

Dla miasta g także mamy $r(g) = 17$. (Jedną z najbardziej odległych od g osad jest osada 6).

Jedynym centrum w tym przykładzie jest miasto f , dla którego $r(f) = 16$. Zatem w tym przykładzie R wynosi 16 .

Usunięcie centrum powoduje podział sieci na kilka spójnych części. Centrum nazwiemy *zrównoważonym*, jeśli każda z tych części zawiera co najwyżej $\lfloor N/2 \rfloor$ osad (zaznaczmy przy tym, że liczymy tylko osady, nie miasta). Przypomnijmy, że $\lfloor x \rfloor$ oznacza największą liczbę całkowitą nie większą niż x .

W naszym przykładzie centrum jest miasto f . Po usunięciu miasta f , sieć zostaje rozbita na cztery spójne części. Części te zawierają następujące zbiory osad: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$ i $\{8, 9\}$. Żadna z części nie zawiera więcej niż $\lfloor 11/2 \rfloor = 5$ osad, zatem miasto f jest zrównoważonym centrum.

Zadanie

Jedyną początkową informacją o sieci miejscowości i drogach jest liczba N osad. Nie znamy liczby miast. Nie wiemy także nic o strukturze połączeń drogowych. Nowe informacje możemy zdobywać zadając tylko pytania o odległości pomiędzy parami osad.

Twoim zadaniem jest:

- We wszystkich podzadaniach: obliczyć odległość R .
- W podzadaniach od 3 do 6: stwierdzić, czy w sieci znajduje się zrównoważone centrum.

Musisz zaimplementować funkcję `hubDistance`. Program sprawdzający rozpatrzy w jednym przebiegu wiele przypadków testowych. Liczba przypadków testowych dla jednego przebiegu nie przekracza **40**. Dla każdego przypadku testowego program sprawdzający wywoła dokładnie raz Twoją funkcję `hubDistance`. Upewnij się, że Twoja funkcja inicjuje wszystkie niezbędne zmienne za każdym razem, gdy jest wywoływana.

- `hubDistance(N, sub)`
 - N : liczba osad.
 - `sub`: numer podzadania (zobacz wyjaśnienie w podrozdziale Podzadania).
 - Jeśli `sub` jest równe 1 lub 2, funkcja może zwrócić R lub $-R$
 - Jeśli `sub` jest większe do 2, to jeśli istnieje zrównoważone centrum funkcja powinna zwrócić R , a w przeciwnym przypadku powinna zwrócić $-R$.

Twoja funkcja `hubDistance` może otrzymywać informacje o sieci dróg wywołując funkcję programu sprawdzającego `getDistance(i, j)`. Ta funkcja zwraca odległość pomiędzy osadami i oraz j . Jeśli i oraz j są takie same, funkcja zwraca **0**. Funkcja zwraca **0** także gdy argumenty są nieprawidłowe.

Podzadania

W każdym przypadku testowym:

- N jest liczbą pomiędzy **6** i **110** włącznie.
- Odległość pomiędzy każdą parą osad wynosi od **1** do **1,000,000** włącznie.

Twój program może wykonać ograniczoną liczbę zapytań. Ograniczenie zależy od podzadania i jest podane w tabeli poniżej. Jeśli Twój program przekroczy dozwoloną liczbę zapytań, to jego wykonywanie zostanie przerwane i wówczas przyjmuje się, że program dał złą odpowiedź.

podzadanie	punkty	liczba zapytań	znajduje zrównoważone centrum	dodatkowe warunki
1	13	$\frac{n(n-1)}{2}$	NIE	brak
2	12	$\lceil \frac{7n}{2} \rceil$	NIE	brak
3	13	$\frac{n(n-1)}{2}$	TAK	brak
4	10	$\lceil \frac{7n}{2} \rceil$	TAK	z każdego miasta wychodzą <i>dokładnie</i> trzy drogi
5	13	$5n$	TAK	brak
6	39	$\lceil \frac{7n}{2} \rceil$	TAK	brak

Przypnijmy, że $\lceil x \rceil$ oznacza najmniejszą liczbę całkowitą większą od lub równą x .

Przykładowy program sprawdzający

Zauważ, że numer podzadania jest częścią danych wejściowych. Przykładowy program sprawdzający zachowuje się różnie w zależności od numeru podzadania.

Przykładowy program sprawdzający czyta dane z pliku `towns.in` podane w następującym formacie:

- wiersz 1: Numer podzadania i liczba przypadków testowych.
- wiersz 2: N_1 , liczba osad w pierwszym przypadku testowym.
- następne N_1 wierszy: j -ta liczba ($1 \leq j \leq N_1$) w i -tym z tych wierszy ($1 \leq i \leq N_1$) jest odległością pomiędzy osadami $i - 1$ oraz $j - 1$.
- Kolejne przypadki testowe, w takim samym formacie jak pierwszy przypadek testowy.

Dla każdego przypadku testowego, przykładowy program sprawdzający wypisuje wartość zwracaną przez `hubDistance` oraz w oddzielnym wierszu liczbę zapytań.

Plik z danymi dla przykładu powyżej ma postać:

```
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
```

Ten format różni się od zwyczajowego specyfikowania listy dróg. Zauważ, że wolno Ci modyfikować przykładowy program sprawdzający, tak żeby używał innego formatu danych.