

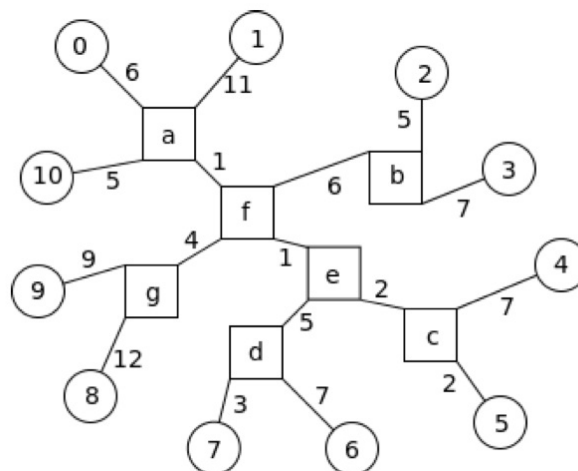
Towns

Er zijn N dorpen in Kazakhstan, genummerd van 0 tot en met $N - 1$. Er is ook een onbekend aantal steden. Dorpen en steden vormen de *gemeentes* van Kazakhstan.

Alle gemeentes van Kazakhstan zijn met elkaar verbonden door één netwerk van bidirectionele snelwegen. Elke snelweg verbindt twee verschillende gemeentes en elk tweetal gemeentes is direct met elkaar verbonden door maximaal één snelweg. Voor elk tweetal gemeentes a en b is er één unieke manier om a te verbinden met b gebruikmakend van snelwegen, zolang geen enkele snelweg meer dan eenmaal gebruikt wordt.

Je weet dat ieder dorp direct verbonden is met één andere gemeente. Iedere stad is direct verbonden met drie of meer gemeentes.

De volgende afbeelding toont een netwerk van **11** dorpen en **7** steden. Dorpen zijn weergegeven als cirkels en zijn gelabeld met integers. Steden zijn weergegeven als vierkanten en zijn gelabeld met letters.



Elke snelweg heeft een positieve integer lengte. De afstand tussen twee gemeentes is de minimale som van de lengte van de snelwegen waarover je moet gaan om van de ene gemeente naar de andere te komen.

Voor elke stad C kunnen we de afstand $r(C)$ meten tot het dorp dat het verst weggelegen is van die stad. Een stad C is een *hub* als de afstand $r(C)$ de kleinste is van alle steden. De afstand tussen een hub en een dorp dat het verste weg ligt van deze hub geven we aan met R . Dus R is de kleinste van alle waarden $r(C)$.

In het gegeven voorbeeld is het verst weggelegen dorp van stad a dorp 8 en de afstand tussen deze twee is $r(a) = 1 + 4 + 12 = 17$. Ook voor stad g geldt $r(g) = 17$. (Eén van de dorpen die het verste weg ligt van stad g is dorp 6 .) De enige hub in het gegeven voorbeeld is stad f met $r(f) = 16$.

. In het gegeven voorbeeld is R dus 16.

Als je een hub verwijdert splitst het netwerk zich in verschillende stukken. Een hub is *gebalanceerd* als ieder van die stukken maximaal $\lfloor N/2 \rfloor$ dorpen bevatten. (Let op: de steden worden dus niet meegeteld!) Met $\lfloor x \rfloor$ bedoelen we de grootste integer die niet groter is dan x .

In ons voorbeeld is stad f een hub. Als we stad f weghalen, dan splitst het netwerk in vier stukken. Deze vier stukken bevatten de volgende verzamelingen van dorpen: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$, and $\{8, 9\}$. Geen van deze stukken heeft meer dan $\lfloor 11/2 \rfloor = 5$ dorpen, dus is stad f een gebalanceerde hub.

Opdracht

De enige informatie die je in het begin over het netwerk van gemeentes en snelwegen hebt is het aantal dorpen N . Je weet niet hoeveel steden er zijn. Je weet niet welke snelwegen aangelegd zijn. Je kunt alleen informatie verkrijgen door vragen te stellen over de afstanden tussen tweetallen dorpen.

Het is jouw taak om te bepalen:

- In alle subtasks: de afstand R .
- In de subtasks 3 tot en met 6: of er wel of geen gebalanceerde hub is in het netwerk.

Je moet de functie `hubDistance` schrijven. De grader evalueert meerdere testcases per run. Het aantal testcases per run is maximaal 40. Voor iedere testcase zal de grader je functie `hubDistance` precies één keer aanroepen. Zorg ervoor dat je functie bij iedere aanroep alle nodige variabelen initialiseert.

- `hubDistance(N, sub)`
 - N : het aantal dorpen.
 - `sub`: het subtask nummer (zie de Subtasks sectie).
 - Als `sub` 1 of 2 is: je functie mag zowel R als $-R$ retourneren.
 - Als `sub` groter is dan 2: als er een gebalanceerde hub is moet je R retourneren, anders retourneer je $-R$.

Je functie `hubDistance` kan informatie over het netwerk van snelwegen verkrijgen door de grader functie `getDistance(i, j)` aan te roepen. Deze functie geeft de afstand terug tussen de dorpen i en j . Als i en j hetzelfde zijn, retourneert de functie 0. Hij geeft ook 0 terug als de argumenten ongeldig zijn.

Subtasks

In iedere testcase:

- N is een getal van 6 tot en met 110.
- De afstand tussen twee verschillende dorpen is een integer van 1 tot met 1 000 000.

Het aantal queries dat je programma mag maken is gelimiteerd. Deze limiet varieert per subtask, zoals gegeven in de tabel hieronder. Als je programma probeert meer queries te doen, dan wordt je programma beëindigd en wordt aangenomen dat je antwoord verkeerd is.

subtask	punten	aantal bevestigingen	gebalanceerde hub bepalen	aanvullende voorwaarden
1	13	$\frac{n(n-1)}{2}$	NEEN	geen
2	12	$\lceil \frac{7n}{2} \rceil$	NEEN	geen
3	13	$\frac{n(n-1)}{2}$	JA	geen
4	10	$\lceil \frac{7n}{2} \rceil$	JA	iedere stad is verbonden met <i>precies</i> drie gemeentes
5	13	$5n$	JA	geen
6	39	$\lceil \frac{7n}{2} \rceil$	JA	geen

Met $\lceil x \rceil$ bedoelen we de kleinste integer die groter dan of gelijk is aan x .

Voorbeeld grader

Let op dat het subtask nummer een onderdeel is van de invoer. De sample grader verandert zijn gedrag afhankelijk van het subtask nummer.

De sample grader leest zijn input uit het bestand `towns.in` op de volgende manier:

- regel 1: Het nummer van de subtask
- regel 2: N_1 is het aantal dorpen in de eerste testcase.
- de volgende N_1 regels: Het j -de getal ($1 \leq j \leq N_1$) op de i -de regel ($1 \leq i \leq N_1$) is de afstand tussen de dorpen $i - 1$ en $j - 1$.
- De volgende testcases die volgen. Deze worden in hetzelfde format gegeven als de eerste testcase.

Voor iedere testcase drukt de grader de returnwaarde af van `hubDistance` en het aantal gemaakt aanroepen af op aparte regels.

Het invoerbestand dat hoort bij het voorbeeld hierboven:

```

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

```

Dit formaat is behoorlijk anders dan een specificatie van de lijst van snelwegen. Let op dat het toegestaan is om de voorbeeld graders aan te passen, zodat je een ander input format kan gebruiken.