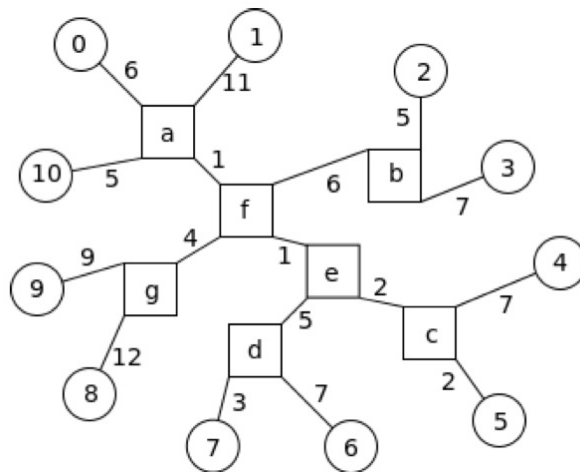


Pueblitos

Hay N pueblitos en Kazajistán, numerados de 0 a $N - 1$. También hay un número desconocido de ciudades. Los pueblitos y ciudades de Kazajistán se denominan en conjunto *asentamientos*.

Todos los asentamientos de Kazajistán están conectados por una sola red de rutas bidireccionales. Cada ruta conecta dos asentamientos distintos. Cada par de asentamientos está conectado directamente por a lo sumo una ruta. Los pueblitos son los asentamientos que están solamente conectados a otro único asentamiento directamente. El resto de los asentamientos son ciudades. Se sabe que no hay ciudades que estén conectadas directamente a exactamente cero o dos asentamientos. En otras palabras, cada ciudad está conectada directamente a tres o más asentamientos. Por último, para cada par de asentamientos a y b hay una manera única en la que se puede pasar de a a b utilizando las rutas, siempre y cuando ninguna ruta se utilice más de una vez.

La siguiente figura muestra una red de **11** pueblitos y **7** ciudades. Los pueblitos se representan como círculos y etiquetados por enteros, las ciudades se representan como cuadrados y etiquetados con letras.



Cada ruta tiene una longitud entera positiva. La distancia entre dos asentamientos es la suma mínima de las longitudes de las rutas que uno necesita para viajar de un asentamiento a otro.

Para cada ciudad C podemos medir la distancia $r(C)$ al pueblito más alejado de esa ciudad. Una ciudad C es un *hub* si la distancia $r(C)$ es la más pequeña entre todas las ciudades. La distancia entre un *hub* y un pueblito que está más alejado del *hub* se denota por R . Por lo tanto, R es el más pequeño de todos los valores de $r(C)$.

En el ejemplo de arriba el pueblito **8** es el más lejano de la ciudad **a**, y la distancia entre ellos es $r(a) = 1 + 4 + 12 = 17$. Para la ciudad **g** también tenemos $r(g) = 17$. (Uno de los pueblitos que está más lejos de **g** es el pueblito **6**). El único *hub* en el ejemplo de arriba es la ciudad **f**, con $r(f) = 16$.

Así que, en el ejemplo de arriba R es **16**.

Quitar un *hub* divide la red en múltiples partes conexas. Un *hub* es *balanceado* si cada una de estas partes contiene a lo más $\lfloor N/2 \rfloor$ pueblitos. (Enfatizamos el hecho de que no contamos las ciudades.). Note que $\lfloor x \rfloor$ denota el entero más grande que no es mayor a x .

En nuestro ejemplo, la ciudad f es un *hub*. Si quitamos la ciudad f , la red se divide en cuatro partes conexas. Estas cuatro partes consisten en los siguientes conjuntos de pueblitos: $\{0, 1, 10\}$, $\{2, 3\}$, $\{4, 5, 6, 7\}$, y $\{8, 9\}$. Ninguna de estas partes tiene más de $\lfloor 11/2 \rfloor = 5$ pueblitos, así que la ciudad f es un *hub* balanceado.

Tarea

Inicialmente, la única información que tiene acerca de la red de asentamientos y rutas es el número N de pueblitos. Usted no tiene el número de ciudades. Tampoco sabe nada acerca de la forma en que están repartidas las rutas en el país. La única manera de obtener información adicional es preguntando por las distancias entre pares de pueblitos.

Su tarea es determinar:

- En toda las subtareas: La distancia R .
- En las subtareas 3, 4, 5 y 6: Si hay un *hub* balanceado en la red.

Usted debe implementar la función `hubDistance`. El evaluador evaluará múltiples casos de prueba en una misma ejecución. El número de casos de prueba por ejecución es a lo sumo **40**. Para cada caso de prueba el evaluador llamará a su función `hubDistance` exactamente una vez. Asegurese que su función inicialice todas las variables necesarias cada vez que sea llamada.

- `hubDistance(N, sub)`
 - N : El número de pueblitos.
 - `sub`: El número de subtarea (explicado en la sección Subtareas).
 - Si `sub` es 1 o 2, la función puede retornar R o $-R$.
 - Si `sub` es mayor que 2, si existe un *hub* balanceado debe retornar R , en caso contrario debe retornar $-R$.

Su función `hubDistance` puede obtener información acerca de la red de las rutas llamando a la función `getDistance(i, j)` del calificador. Esta función retorna la distancia entre los pueblitos i y j . Note que si i y j son iguales, la función retorna **0**. También se retorna **0** si los argumentos son inválidos.

Subtareas

En cada caso de prueba:

- N está entre **6** y **110**, inclusive.
- La distancia entre cualquier par de pueblitos está entre 1 y 1,000,000, incluidos.

El número de preguntas que su programa puede hacer es limitado. El límite varía de acuerdo a la subtarea, como se muestra en la tabla de abajo. Si su programa trata de exceder el límite de preguntas, será terminada su ejecución y se toma como una respuesta incorrecta.

subtarea	puntos	número de preguntas	encontrar hub balanceado	restricciones adicionales
1	13	$\frac{N(N-1)}{2}$	NO	ninguna
2	12	$\lceil \frac{7N}{2} \rceil$	NO	ninguna
3	13	$\frac{N(N-1)}{2}$	SI	ninguna
4	10	$\lceil \frac{7N}{2} \rceil$	SI	cada ciudad está conectada a <i>exactamente</i> tres asentamientos
5	13	$5N$	SI	ninguna
6	39	$\lceil \frac{7N}{2} \rceil$	SI	ninguna

Note que $\lceil x \rceil$ denota el entero mas pequeño que es mayor o igual a x .

Evaluador de ejemplo

Note que el número de subtarea es parte de la entrada. El evaluador de ejemplo cambia su comportamiento de acuerdo al número de subtarea.

El evaluador de ejemplo lee la entrada del archivo `towns.in` en el siguiente formato:

- línea 1: El número de subtarea y el número de casos de prueba.
- línea 2: N_1 , el número de pueblitos en el primer caso de prueba.
- Sigüientes N_1 líneas: El j -ésimo número ($1 \leq j \leq N_1$) en la i -ésima de estas líneas ($1 \leq i \leq N_1$) es la distancia entre los pueblitos $i - 1$ y $j - 1$.
- Los sigüientes casos ejemplo van a continuación. Son dados en el mismo formato que el primero.

Para cada caso de prueba, el evaluador de ejemplo imprime el valor de retorno de `hubDistance` y el número de llamadas en una línea separada.

Este formato es bastante diferente de especificar la lista de carreteras. Note que está permitido modificar el calificador de ejemplo para que pueda usar un formato de entrada distinto.

La entrada correspondiente al ejemplo de arriba es:

```
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
```

Este formato es bastante diferente de especificar la lista de rutas. Note que está permitido modificar el evaluador de ejemplo para que pueda usar un formato de entrada distinto.