

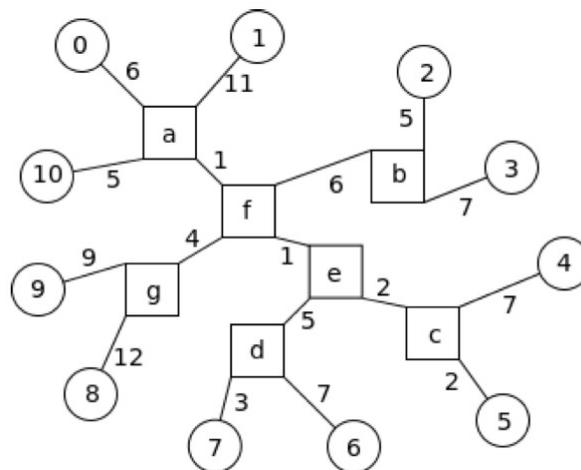
# Pueblos

En Kazajistán existen  $N$  pueblos pequeños, enumerados desde  $0$  hasta  $N - 1$ . También existe un número desconocido de ciudades grandes. Los pueblos pequeños y las ciudades grandes de Kazajistán son conjuntamente llamados *asentamientos*.

Todos los asentamientos de Kazajistán están conectados por una red de bidireccional de autopistas. Cada autopista conecta dos asentamientos distintos, y cada par de asentamientos está directamente conectado por a lo sumo una autopista. Para cada par de asentamientos  $a$  y  $b$  hay una única manera para ir de  $a$  a  $b$  utilizando las autopistas, siempre y cuando ninguna autopista se use más de una vez.

Se conoce que cada pueblo pequeño está directamente conectado a un único otro asentamiento, y cada ciudad grande está directamente conectada a tres o más asentamientos.

La siguiente figura muestra una red de **11** pueblos pequeños y **7** ciudades grandes. Los pueblos pequeños están representados con círculos y etiquetados con enteros, mientras que las ciudades grandes están representadas con cuadrados y etiquetadas con letras.



Cada autopista posee una longitud entera y positiva. La distancia entre dos asentamientos es la mínima suma de las longitudes de las autopistas que se deben atravesar para llegar de un asentamiento a otro.

Para cada ciudad grande  $C$  podemos medir la distancia  $r(C)$  hasta el pueblo más lejano de esa ciudad. Una ciudad grande  $C$  es un *hub* si la distancia  $r(C)$  es la menor entre todas las ciudades. La distancia entre un *hub* y un pueblo que sea el más lejano del hub será denotado por  $R$ . Entonces,  $R$  es el menor de todos los valores  $r(C)$ .

En el ejemplo anterior el pueblo más lejano desde la ciudad  $a$  es el pueblo  $8$ , y la distancia entre ellos es  $r(a) = 1 + 4 + 12 = 17$ . Para la ciudad  $g$  también tenemos  $r(g) = 17$ . (Uno de los pueblos que está más alejados de  $g$  es el pueblo  $6$ .) El único hub en el ejemplo anterior es la ciudad  $f$ , con  $r(f) = 16$ . Por lo tanto, en el ejemplo anterior  $R$  es  $16$ .

Remover un hub divide la red en múltiples partes conectadas. Un hub está *balanceado* si cada una de las partes contiene a lo sumo  $\lfloor N/2 \rfloor$  pueblos pequeños. (Hacemos énfasis en que no se cuentan las ciudades grandes). Nótese que  $\lfloor x \rfloor$  denota el entero más grande que no es mayor que  $x$ .

En nuestro ejemplo, la ciudad  $f$  es un hub. Si removemos la ciudad  $f$ , la red se dividirá en cuatro partes conectadas. Estas cuatro partes consisten en los siguientes conjuntos de pueblos pequeños:  $\{0, 1, 10\}$ ,  $\{2, 3\}$ ,  $\{4, 5, 6, 7\}$ , y  $\{8, 9\}$ . Ninguna de las partes tiene más de  $\lfloor 11/2 \rfloor = 5$  pueblos pequeños, por lo tanto la ciudad  $f$  es un hub balanceado.

## Tarea

Inicialmente, la única información que tienes acerca de la red de asentamientos y autopistas es el número  $N$  de pueblos pequeños. No conoces el número de ciudades grandes. Tampoco conoces nada sobre la disposición de las autopistas en el país. Sólo puedes obtener nueva información haciendo consultas sobre las distancias entre pares de pueblos pequeños.

Tu tarea es determinar:

- En todas las sub-tareas: la distancia  $R$ .
- En las sub-tareas 3 a la 6: si existe en la red un hub balanceado.

Necesitas implementar la función `hubDistance`. El grader evaluará múltiples casos de prueba en una misma ejecución. El número de casos de prueba por cada corrida será como máximo **40**. Para cada caso de prueba el grader llamará la función `hubDistance` exactamente una sola vez. Asegúrate que tu función inicialice todas las variables necesarias cada vez que sea llamada.

- `hubDistance(N, sub)`
  - $N$ : el número de pueblos pequeños.
  - `sub`: el número de sub-tarea (explicado en la sección Sub-tareas)
  - Si `sub` es 1 o 2, la función podrá retornar  $R$  o  $-R$
  - Si `sub` es mayor que 2, si existe un hub balanceado entonces la función debe retornar  $R$ , de lo contrario debe retornar  $-R$ .

Tu función `hubDistance` puede obtener información acerca de la red de autopistas llamando la función del grader `getDistance(i, j)`. Esta función retorna la distancia entre los pueblos pequeños  $i$  y  $j$ . Note que si  $i$  y  $j$  son iguales, la función retornará **0**. También retornará **0** cuando los argumentos sean inválidos.

## Sub-tareas

En cada caso de prueba:

- $N$  estará entre **6** y **110** inclusive.
- La distancia entre dos pueblos distintos cualesquiera estará entre 1 y 1,000,000 inclusive.

El número de consultas que tu programa puede hacer es limitado. El límite varía por sub-tarea, tal y como se especifica en la tabla siguiente. Si tu programa intenta exceder el límite del número de consultas, será terminado y se asumirá que ha dado una respuesta incorrecta.

sub-tarea	puntos	número de consultas	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 autopistas
5	13	$5n$	SI	ninguna
6	39	$\lceil \frac{7n}{2} \rceil$	SI	ninguna

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

## Grader de Ejemplo

Nota que el número de la sub-tarea es parte de la entrada. El grader de ejemplo cambiará su comportamiento de acuerdo con el número de la sub-tarea.

El grader de ejemplo leerá el archivo de entrada `towns.in` en el siguiente formato:

- línea 1: Número de la sub-tarea y número de casos de prueba.
- línea 2:  $N_1$ , el número de pueblos pequeños en el primer caso de prueba.
- las siguientes  $N_1$  líneas: El  $j$ -ésimo ( $1 \leq j \leq N_1$ ) en el  $i$ -ésimo de estas líneas ( $1 \leq i \leq N_1$ ) es la distancia entre los pueblos pequeños  $i - 1$  y  $j - 1$ .
- Siguen los próximos casos de prueba. Serán dados en el mismo formato que el primer caso de prueba.

Para caso de prueba, el grader de ejemplo imprimirá el valor de retorno de `hubDistance` y en una línea separada el número de llamadas realizadas.

El archivo de entrada correspondiente para el ejemplo anterior 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 muy distinto a especificar la lista de autopistas. Nota que se te es permitido modificar a los graders de ejemplo, tal que puedan usar distintos formatos de entrada.