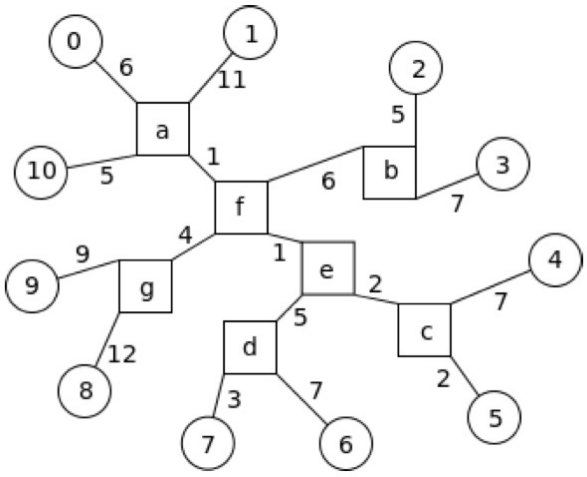
Towns

Hay N pequeños pueblos en Kazajistán, numerados desde 0 hasta $N-1$. También hay un número desconocido de ciudades grandes. Los pueblos pequeños y las ciudades grandes son llamados en conjunto *asentamientos*.

Todos los asentamientos de Kazajistán son conectados por una red de autopistas bidireccionales. Cada autopista conecta dos asentamientos diferentes y cada par de asentamientos es conectado directamente por a lo sumo una autopista. Para cada par de asentamientos a y b hay un camino único para ir de a a b usando las autopistas, siempre y cuando no se use ninguna autopista más de una vez.

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

La siguiente figura muestra una red de 11 pueblos pequeños y 7 ciudades grandes. Los pueblos pequeños se muestran como círculos y son etiquetados con enteros, las ciudades grandes son mostradas como cuadrados y etiquetadas con letras.



Cada autopista tiene un entero positivo que representa su longitud. La distancia entre dos asentamientos es la mínima suma de la longitudes de las autopistas que uno necesita recorrer para ir de un asentamiento al otro.

Para cada ciudad grande C podemos medir la distancia $r(C)$ al pueblo pequeño que está mas lejos de esa ciudad. Una ciudad grande C es un *eje* si la distancia $r(C)$ es la más pequeña entre todas las ciudades grandes. La distancia entre un eje y un pueblo pequeño que está lo más alejado del eje se denota por R. Así, R es el más pequeño de los valores $r(C)$.

En el ejemplo anterior el pueblo pequeño más lejano de 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 que $r(g) = 17$. (Uno de los pueblos pequeños que está más alejado de g es el pueblo 6.) El único eje en el ejemplo anterior es la

ciudad f, con $r(f) = 16$. Entonces, en el ejemplo anterior R es 16.

Eliminar un eje divide la red en múltiples partes conectadas. Un eje está *balanceado* si cada una de esas partes contiene a lo sumo $\lfloor N/2 \rfloor$ pueblos pequeños. (Hacemos incapié en que no contamos 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 eje. Si eliminamos la ciudad f, la red se dividirá en cuatro partes conectadas. Esas cuatro partes están formadas por los siguientes conjuntos de pueblos pequeños: { $0, 1, 10$}, {$2, 3$}, {$4, 5, 6, 7$}, y {$8, 9$}. Ninguna de esas partes tiene más de $\lfloor 11/2 \rfloor = 5$ pueblos pequeños, por lo tanto la ciudad f es un eje balanceado.

Task

Inicialmente la única información que usted tiene acerca de la red de asentamientos y autopistas es el número N de pueblos pequeños. Usted no sabe el número de ciudades grandes. Usted tampoco sabe nada acerca de la distribución de autopistas en el país. Usted solo puede obtener nueva información haciendo consultas acerca de las distancias entre pares de pueblos pequeños.

Su tarea es determinar:

- En todas las subtareas: la distancia R.
- En las subtareas 3 a la 6: si hay un eje balanceado en la red

Usted debe implementar la función `hubDistance`. El grader evaluará múltiples casos de prueba en una sola corrida. El número de casos de prueba es a lo sumo 40. Para cada caso de prueba el grader llamará su función `hubDistance` exáctamente una vez. Asegúrese de que su función inicializa todas las variables necesarias cada vez que es llamada.

- `hubDistance(N, sub)`
  - `N`: el número de pueblos pequeños.
  - `sub`: el número de subtarea (explicado en la sección Subtasks).
  - Si `sub` es 1 o 2, la función puede retornar R o $-R$
  - Si `sub` es mayor que 2, Si existe un eje balanceado entonces la función debe retornar R, de lo contrario $-R$.

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

Subtasks

En cada caso de prueba:

- N está entre 6 y 110 inclusive.
- La distancia entre cualesquiera dos pueblos pequeños está entre 1 y 1.000.000 inclusive.

El número de consultas que su programa puede hacer es limitado. El límite varía por subtarea, como aparece en la siguiente tabla. Si su programa trata de exceder el límite de consultas, será terminado y se asumirá que dió una respuesta incorrecta.

subtarea	puntos	número de consultas	encontrar eje 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 grande está conectada a *exáctamente* tres asentamientos
5	13	$5N$	SI	ninguna
6	39	$\lceil \frac{7N}{2} \rceil$	SI	ninguna

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

Implementation details

Usted debe enviar exáctamente un archivo, llamado `towns.c`, `towns.cpp`, `towns.pas` o `towns.java`. Este archivo debe implementar el programa descrito anteriormente, usando las siguiente firmas. Además debe incluir el archivo cabecera `towns.h` para la implementación C/C++.

Este archivo implementa los subprogramas escritos anteriormente usando las siguientes firmas.

C/C++ program (incluya `towns.h` al principio del archivo fuente)

```
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);
```

Sample grader

Nótese que el número de subtarea es parte de la entrada. El grader de ejemplo cambia su comportamiento acorde al número de subtarea.

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

- línea 1: Número de subtarea y el 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 número $(1 \leq j \leq N_1)$ en la i-ésima de estas líneas

$(1 \leq i \leq N_1)$ es la distancia entre entre los pueblos pequeños $i-1$ y $j-1$.

- Los siguientes casos de prueba siguen. Estos son dados en el mismo formato que el primer caso de prueba.

Para cada caso de prueba, el grader de ejemplo imprime el valor de retorno de `hubDistance` y el número de llamadas hechas en líneas separadas.

For each test case, the sample grader prints the return value of `hubDistance` and the number of calls made on separate lines.

El archivo de entrada que corresponde al ejemplo anterior es:

The input file corresponding to the example above is:

```
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 algo diferente de espeificar la lista de autopistas. Usted está abilitado a modificar los graders de ejemplo, para que usen un formato de entrada diferente.