

Recuperando distancias

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

En un país lejano, existen N pueblos ubicados sobre una misma ruta. En un extremo de esta ruta se ubica una central de correo, encargada de planear y coordinar los envíos de correspondencia entre los distintos pueblos. Cada pueblo se encuentra a una cierta distancia en km de esta central, y denotamos estas distancias $0 < d_1 < d_2 < \dots < d_N$.



En la central de correo se guarda una valiosa tabla con todos los pares de distancias entre los distintos pueblos y la central, que es útil para calcular los tiempos de viaje de los correos. Por ejemplo, una tabla para el caso del dibujo podría ser la siguiente:

Central - Pueblo 1 : 2km

Pueblo 1 - Pueblo 2 : 1km

Pueblo 1 - Pueblo 3 : 8km

Central - Pueblo 3 : 10km

Central - Pueblo 2 : 3km

Pueblo 2 - Pueblo 3 : 7km

Notar que esta tabla siempre tiene $\frac{N(N+1)}{2}$ entradas, ya que tiene todos los pares posibles.

Carlos es un joven empleado en la central, que tiene a su cargo entre otras cosas el cuidado de esta tabla. El problema es que Carlos es muy despistado, y en un descuido tiró el papel con la tabla a la basura, pensando que era el envoltorio de las medialunas del desayuno.

Por suerte, Carlos tiene guardada en el cajón del mate una copia de seguridad que hizo de la tabla. El problema es que en la copia anotó solamente las distancias ¡Sin indicar los pueblos correspondientes! De esta manera, la copia de seguridad de Carlos para el dibujo podría ser:

2km-1km-8km-10km-3km-7km

Carlos está en problemas, ya que como solamente tiene anotadas las $\frac{N(N+1)}{2}$ distancias de la tabla original, pero estas no están en ningún orden en particular, no tiene idea de cuál es cada una.

Para evitar que sus superiores se enfaden con él, Carlos necesita tu ayuda. Debes escribir un programa que calcule **todas** las posibilidades existentes para las N distancias originales, dada la tabla final que registró Carlos. Por ejemplo, para los 6 valores anteriores existen únicamente 2 posibilidades:

$d_1 = 2, d_2 = 3, d_3 = 10$

o bien

$d_1 = 7, d_2 = 8, d_3 = 10$

Entrada y salida

En este problema se deben entregar **únicamente los archivos de salida** (output-only), y no los programas (en caso de usar) con los cuales fueron obtenidos.

Cada **archivo de entrada** contiene:

- Primera línea: Un entero N , con la cantidad de pueblos.
- Segunda línea: $\frac{N(N+1)}{2}$ enteros, separados por espacios, con todas las distancias correspondientes entre los pueblos. Podrían estar en cualquier orden.

Cada **archivo de salida** debería contener una línea por cada solución posible, indicando los correspondientes valores d_1, d_2, \dots, d_N , separados por un espacio. Estas líneas pueden estar en cualquier orden.

Ejemplo

El ejemplo anterior correspondería al siguiente archivo de entrada:

```

3
2 1 8 10 3 7
```

Y una posible salida correcta sería:

```

7 8 10
2 3 10
```

Archivos y puntuación

Se reciben 10 archivos de entrada (numerados de 1 a 10). Cada uno de ellos tendrá un caso a resolver.

Por cada caso de prueba:

- Si no se entrega un archivo de salida para ese caso, o si la salida tiene 0 líneas (es un archivo vacío), se reciben 0 puntos.
- Si alguna de las líneas del archivo de salida no es una solución correcta, se reciben 0 puntos.
- Si hay más de una línea con la misma solución en el archivo de salida, se reciben 0 puntos.
- Si no se dan las anteriores, pero el archivo no contiene **TODAS** las soluciones, se reciben $3+4 \cdot \frac{e}{T}$ puntos, donde e es la cantidad de soluciones en el archivo, y T la cantidad total de soluciones que existen.
- Si el archivo contiene todas las soluciones correctas, se obtienen 10 puntos.

Notar que cada caso tiene entonces un puntaje de 0 a 10, con lo cual los puntajes totales para el problema van desde 0 hasta 100 puntos.

Subtareas

- En el caso 1, $N = 5$.
- En el caso 2, $N = 14$ y existe una solución en la cual **exactamente uno** de los d_i es par.
- En los casos 3 a 8, $N \leq 30$, $d_i \leq 10^{18}$
- En los casos 9 y 10, $N = 100$, $d_i \leq 106$

Además, notar que en un problema output-only, se conocen **todos** los casos, y cada uno puede tener sus particularidades más allá de lo indicado. Ver los archivos de entrada para conocer los casos exactos a resolver.