

El Jardín de Fumito

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Fumito posee un enorme y hermoso jardín en las afueras de Kyoto. Existen en la zona N postes, numerados desde 0 hasta $N - 1$ inclusive. Cada uno de ellos se ubica en una cierta posición (x_i, y_i) en el plano. **No existen 3 postes alineados.**

Las vallas que lo delimitan forman un polígono simple, con vértices en los N postes. La i -ésima valla (numerando desde 0) va desde el poste i hasta el $i + 1$, excepto la valla $N - 1$ -ésima que va desde el poste $N - 1$ hasta el 0 . Las vallas se corresponden así a segmentos de recta en el plano, y dos vallas distintas nunca se intersecan excepto en sus extremos. Los postes están numerados en sentido antihorario en el plano, es decir: si se camina por la valla i , desde el poste i hasta el $i + 1$ (o desde el $N - 1$ hasta el 0 para la última valla), el interior del polígono queda a la izquierda de la valla.

Fumito es una persona extremadamente supersticiosa. Está completamente convencido de que hay una fuente de "energía negativa", que irradia desde un único punto del plano que está **estrictamente dentro** de su jardín. Su preocupación lo ha llevado a contratar a un renombrado grupo de hechiceros.

Los hechiceros afirman ser capaces de destruir la fuente de energía negativa mediante un complicado ritual de *invocación* de espíritus. El ritual es llevado a cabo por exactamente **3** hechiceros, que se colocan cada uno en algún punto distinto del plano y recitan las palabras mágicas correspondientes. El ritual destruye por completo cualquier fuente de energía negativa que hubiera estrictamente dentro del triángulo que tiene a los **3** hechiceros en sus vértices.

Los vecinos de Fumito se enojarán con

él si ven a un hechicero tan cerca de sus propiedades: para evitar este inconveniente, Fumito ha pedido a los hechiceros que se disfracen de trabajadores de reparación de postes. Por esta razón, **todo ritual de invocación de espíritus debe llevarse a cabo con los hechiceros ubicados en algunos de los N postes.**

Como este ritual es sumamente agotador, debe realizarse **exactamente una vez**. Los hechiceros utilizarán antes, tantas veces como sea necesario, un ritual diferente que les permite "sentir" aproximadamente desde qué dirección viene la energía negativa.

Para realizar este ritual, exactamente dos hechiceros se colocan cada uno en un poste diferente. Uno de ellos tomará el rol de *hechicero principal*, y el otro el de *hechicero secundario*. **El segmento entre ambos hechiceros debe estar completamente contenido en el jardín.** Este segmento divide al jardín en exactamente dos regiones conexas.

Cuando el hechicero principal mira hacia el hechicero secundario, una de estas regiones está a su izquierda, y la otra a su derecha. Realizando este ritual, el hechicero principal descubre desde cuál de estas dos regiones proviene la energía negativa, por lo que estará seguro de que la fuente está en esa región.

Si la fuente de energía negativa estuviera en el segmento que une ambos hechiceros, serían capaces de detectarlo, pero Fumito está completamente seguro de que esto nunca ocurrirá, ya que se siente realmente desafortunado. Es decir, Fumito está seguro de que **la fuente de energía negativa no está exactamente entre dos postes.**

Fumito le pagará a los hechiceros un monto proporcional a la cantidad total

de rituales que realicen, por lo que está interesado en minimizar esta cantidad. Debes escribir una función que interactúe con los hechiceros, para encontrar la fuente de energía negativa sin realizar demasiados rituales.

Descripción de la función

Debes implementar la función `fumito(x,y)`, que recibe dos arreglos de **N** enteros. El *i*-ésimo poste se encuentra ubicado en $(x[i], y[i])$.

En su implementación podrás llamar a las siguientes funciones:

- `sentir(postePrincip, posteSecun)`.
Esta función se utiliza para realizar un ritual que permite descubrir desde dónde viene la energía negativa. Se deben indicar como parámetros los índices de los postes donde se ubicará cada hechicero. La función retorna 0, si la energía negativa viene desde la **izquierda** (de acuerdo a la vista del hechicero principal), y retorna 1 en caso contrario.
- `destruir(poste1, poste2, poste3)`.
Esta función se debe llamar una única vez al final, para realizar el ritual de invocación de espíritus que destruirá la fuente de energía negativa. Se deben indicar los índices de los postes en los que se ubicarán los hechiceros para este ritual.

Se puede realizar un **máximo de 20 llamadas** a estas funciones (incluyendo la llamada final a `destruir`).

Evaluador

Para poder probar su programa interactivamente, dispone de un evaluador local. Este lee de la entrada estándar con el siguiente formato:

- Una línea con un entero **N**
- **N** líneas, cada una con dos enteros x_i, y_i

El evaluador ejecutará la función `fumito` con el polígono leído, y mostrará por pantalla las llamadas a las funciones `sentir` y `destruir`. En el caso de las llamadas a `sentir`, el evaluador leerá la respuesta de la entrada estándar, y retornará ese valor leído.

Restricciones

- $3 \leq N \leq 2.500$
- $-10^9 \leq x_i, y_i \leq 10^9$

Ejemplo

Si se ejecuta el evaluador con la siguiente entrada:

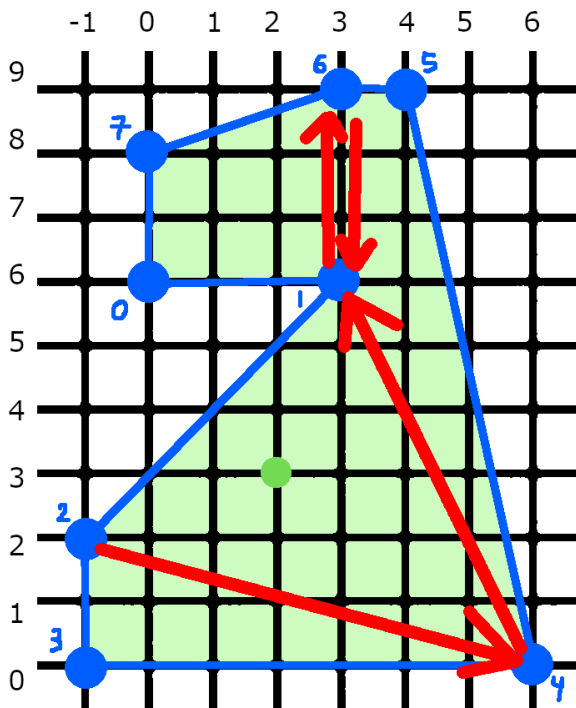
```
8
0 6
3 6
-1 2
-1 0
6 0
4 9
3 9
0 8
```

Una posible interacción sería:

```
sentir(6, 1)
0
sentir(1, 6)
1
sentir(2, 4)
0
sentir(4, 1)
0
destruir(2, 1, 4)
```

Notar que es el participante al usar el evaluador quien ha dado las respuestas **0, 1, 0, 0**, que son correctas por ejemplo para el caso en que la fuente de energía se encuentre en **(2, 3)**

En la siguiente figura, los postes y vallas están en azul, y flechas rojas indican las llamadas a `sentir`, dirigidas del hechicero principal al secundario.



Puntuación

En cada ejecución, el programa deberá destruir exitosamente la fuente de energía negativa. Si el programa no llama a la función `destruir`, la llama más de una vez, realiza más llamadas luego de llamar a `destruir`, o si es posible de acuerdo a las llamadas a `sentir` realizadas que la fuente esté fuera del triángulo indicado, entonces el programa obtendrá 0 puntos para la subtarea.

También se obtienen cero puntos si se llama a alguna función con parámetros inválidos (fuera de rango, postes repetidos, el segmento corta al polígono, etc), o si se realizan más de **20** llamadas en total, a las funciones `sentir` y `destruir`.

Notar que el corrector de este problema es adaptativo: la ubicación de la fuente de energía negativa puede no estar prefijada, sino ajustarse durante la ejecución de acuerdo a las llamadas que realice el programa. Las respuestas retornadas por la función `sentir` siempre serán consistentes con alguna ubicación válida para la fuente de energía.

Subtareas

1. El polígono es convexo (9 puntos)
2. $N \leq 20$ (12 puntos)
3. $N \leq 100$ (15 puntos)
4. $N \leq 500$ (15 puntos)
5. Sin más restricción (49 puntos)