

## Encontrando el Tiraníc

Contribución de Agustín Santiago Gutiérrez y Facundo Gutiérrez

### Descripción del problema

Hace 105 años, el SRM Tiraníc naufragó al chocar con un iceberg. Se tiene localizada la zona del naufragio, pero todavía resta localizar la posición exacta de la embarcación.

La marina logró rodear la zona, que se encuentra dividida en  $m \times n$  casillas formando una grilla cuadrículada orientada siguiendo paralelos y meridianos, y se sabe que el barco tiene forma de rectángulo, de área desconocida. Además, la marina dispone de una serie de radares en los bordes de la grilla, con los que es posible realizar mediciones para encontrar el navío.

Para utilizar un radar, se debe:

1. Indicar desde que borde de grilla: N, E, S u O se va a sondear.
2. Indicar el número de fila o columna a sondear.

Hecho esto, el radar barre toda una fila o columna completa desde la dirección indicada, y da a conocer si en esa región barrida se detectaron vestigios de la embarcación, y además de ser así, a qué distancia del radar se encuentra.

Por ejemplo la siguiente sería una grilla de  $6 \times 9$  con su Norte en la parte superior, donde las X muestran una hipotética ubicación del Tiraníc.

```

.....
..XXXXXXX
..XXXXXXX
.....
.....
.....
.....

```

Si así fuera y se preguntara **Oeste**, fila **1**, el radar devolvería que no hubo detección.

Lo mismo si se preguntara desde el **Este**. Si se pregunta por las filas **2** o **3** en cambio, al preguntar desde el **Oeste** el radar indicaría una detección a distancia **2**, y desde el **Este** una detección a distancia **0**.

Similarmente, si se pregunta desde el **Norte** en columna **3**, se obtiene detección a distancia **1**, y desde el **Sur**, detección a distancia **3**. Al preguntar por las columnas **1** y **2** no se obtiene detección en ninguna de las dos direcciones.

El objetivo de la marina es encontrar el rectángulo que representa al SRM Tiraníc, haciendo la menor cantidad de mediciones

posibles (pues cada medición de radar utiliza una enorme cantidad de energía).

### Restricciones

- Los lados del barco son paralelos a los de la grilla, cuya área será como máximo **1.000.000**
- Una casilla de la grilla pertenece íntegramente o al barco o al mar circundante por cuanto los 4 ángulos de las esquinas del barco están en puntos de la grilla
- Por las descripciones históricas disponibles, se sabe que el área del Tiraníc ( medida en casillas) es mayor o igual que **100**.

### Tarea e interacción

Este es un problema interactivo: Se debe proveer una subrutina **tiraníc**( **m**, **n** ENTEROS; **distancias** ARREGLO[4] de ENTEROS ) siendo sus parámetros:

**m n**: las dimensiones de la grilla anteriormente mencionadas. Ambas son enteros positivos.

**distancias**: vector para retornar la ubicación del tiraníc. Expresan en casillas, las distancias a los bordes Norte, Este, Sur y Oeste de la grilla. Para el ejemplo anteriormente expuesto, se recibirían **m=6** y **n=9**, y se deberían devolver los siguientes valores: **1, 0, 3 y 2**.

En la implementación de la subrutina **tiraníc**, se pueden realizar llamadas a una función **radar**( **borde** CARACTER, **i** ENTERO ) que retorna un ENTERO, que permite realizar mediciones de radar y toma los siguientes parámetros:

**borde**: Se debe indicar una de las cuatro letras **N, E, S u O**.

**i**: Se debe indicar el número de fila o columna por la cual se pregunta. Tanto filas como columnas se numeran desde **1**, las filas de Norte hacia Sur y las columnas de Oeste a Este.

La función radar devuelve la distancia medida hasta el barco, expresada en casillas, o bien **-1** para indicar que no se ha detectado nada en esa fila/columna.

La cantidad total de llamadas a la función radar está limitada (ver tabla al final del texto).

### Ejemplos de interacción

El cuadro adjunto muestra las llamadas válidas a la función radar mencionadas al desarrollar el ejemplo, junto con el resultado que se

Llamada	Resultado
radar( 'O', 1 )	-1
radar( 'E', 1 )	-1
radar( 'O', 2 )	2
radar( 'O', 3 )	2
radar( 'E', 2 )	0
radar( 'E', 3 )	0
radar( 'N', 3 )	1
radar( 'S', 3 )	3
radar( 'N', 2 )	-1
radar( 'S', 2 )	-1
radar( 'N', 1 )	-1
radar( 'S', 1 )	-1

obtuvo.

### Evaluador local

El evaluador local lee los datos de entrada por consola en el siguiente orden:

- Una línea con dos números m y n.
- Una línea con 4 números, indicando los datos del tiranico: las 4 distancias a los 4 bordes, en el orden **N E S O**.

El evaluador local no valida las restricciones de tamaño de la grilla y del barco pero si verifica que los 6 números representen una posible grilla y dentro de ella un posible barco.

Entrega por consola la cantidad de mediciones realizadas, y los datos del barco devueltos por el subprograma.

Por ejemplo, si se ingresara:

```
#el ejemplo
6 9
1 0 3 2
```

Con un programa correcto que realizara las mediciones anteriormente mencionadas, se obtendría por pantalla:

```
Se realizaron 12 mediciones.
Distancias al barco: 1 0 3 2
```

### Evaluación del programa

Su programa será evaluado con un corrector adaptativo, es decir, la posición del barco podría no estar prefijada al comienzo de la ejecución del programa, sino que puede depender de las llamadas que la subrutina **tiranico** realice a la función **radar**.

Se garantiza, sin embargo, que todas las respuestas que se obtendrán como resultado de la función **radar** serán siempre consistentes con alguna ubicación posible del barco.

Para obtener los puntos correspondientes a un cierto caso de prueba, su programa deberá identificar correctamente todos los datos del barco para ese caso. De ser así, el porcentaje del puntaje total que se obtendrá dependerá de la cantidad de llamadas a la función **radar** que el programa haya realizado, de acuerdo a la tabla que se muestra a continuación.

Si se realiza alguna llamada inválida a la función **radar**, se obtienen cero puntos en ese caso de prueba.

Máxima cantidad de mediciones de radar permitidas	Porcentaje de los puntos totales
<b>1005</b>	<b>5</b>
<b>700</b>	<b>10</b>
<b>340</b>	<b>25</b>
<b>310</b>	<b>50</b>
<b>190</b>	<b>100</b>