XXX Olimpiada Informática Española

OIE 2026 — Final Día 1

11 de abril de 2026

Organizan y financian

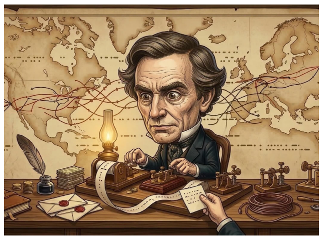
Patrocinan

Listado de problemas

Autores y revisores de los problemas: María Lucía Aparicio, Max Balsells, Izan Beltran, Hugo Domínguez, Blanca Huergo, Roger Lidón, Huize Mao, Darío Martínez, Edgar Moreno, Daniel Nieto, Pablo Sáez, Manuel Torres, Anier Velasco, Alberto Verdejo, Alejandro Vivero.

Puntos y rayas

MORSE



El código morse es un sistema de representación de caracteres mediante señales cortas y largas, tradicionalmente llamadas *puntos* (`.`) y *rayas* (`-`). Fue desarrollado en el siglo XIX para su uso en el telégrafo y permitió transmitir información a grandes distancias utilizando secuencias de impulsos.

En la versión estándar del código morse, cada dígito decimal del 0 al 9 se representa mediante una combinación de exactamente cinco símbolos, según la siguiente tabla:

Dígito	**Código morse**		**Dígito**	**Código morse**
0	`-----`		5	`.....`
1	`.----`		6	`-....`
2	`..---`		7	`--...`
3	`...--`		8	`---..`
4	`....-`		9	`----.`

Para codificar una secuencia de dígitos, se concatenan las representaciones correspondientes a cada uno de ellos, sin añadir separadores adicionales.

Dada una secuencia de dígitos, queremos saber cuántos puntos y cuántas rayas hay en total en su codificación en código morse.

Entrada y salida

La entrada comienza con un entero T, que indica el número de casos de prueba.

Cada uno de los siguientes T casos consiste en una línea con una secuencia no vacía s de dígitos decimales (`0`–`9`).

Para cada caso de prueba, se debe imprimir una línea con dos enteros separados por un espacio: el número total de puntos seguido del número total de rayas al codificar en código morse la secuencia s.

Restricciones

- $1 \leq T \leq 1000$
- La longitud de la secuencia s varía entre 1 y 1000.

Subtareas

1. (15 puntos) En la secuencia s solamente se usa el dígito `0`.
2. (20 puntos) La longitud de la secuencia s es 1.
3. (25 puntos) Todos los dígitos de la secuencia s son iguales.
4. (40 puntos) Sin restricciones adicionales.

Ejemplo

Entrada	Salida
3 5 10 314159	5 0 1 9 15 15

Canal con interferencias

OPERACION

El Profesor Oak ha enviado un mensaje a través de un canal con demasiadas interferencias. Este mensaje consiste en una secuencia s de n *bits* (0 o 1), $s_1\ s_2 \ldots s_n$, que tardará k milisegundos en llegar a su destinatario. Debido a estas interferencias, cada milisegundo el mensaje sufre una transformación de forma que s pasa a ser s', donde $s'_i = s_{i-1} \oplus s_{i+1}$†. Aquí los índices se toman de manera cíclica, es decir, $s_0 = s_n$ y $s_{n+1} = s_1$.

¿Puedes anticipar cuál será el mensaje final que le llegará al destinatario después de k milisegundos?

Entrada y salida

La primera línea de la entrada contiene el número de casos T.

Cada caso ocupa dos líneas. En la primera aparecen dos enteros, n y k, la longitud de la secuencia s y los milisegundos que tarda en llegar el destinatario, respectivamente. En la segunda línea aparece la secuencia s de n bits.

Para cada caso debes imprimir la transformación de s tras k milisegundos de interferencias.

Restricciones

- $1 \leq T \leq 3 \cdot 10^4$
- $3 \leq n \leq 10^5$
- La suma de n para todos los casos es como mucho 10^5.
- $1 \leq k \leq 10^{18}$
- $s_i \in \{0, 1\}$

Subtareas

1. (13 puntos) Tanto la suma de n como la suma de k para todos los casos son, como máximo, 1000.
2. (28 puntos) $k \geq 10^{15}$, n es potencia de 2 ($n = 2^r$ para algún r entero).
3. (22 puntos) La suma de n para todos los casos es como mucho 1000.
4. (37 puntos) Sin restricciones adicionales.

Ejemplo

Entrada

```
3
7 1
1001111
3 1000
101
5 2
11011
```

Salida

```
1111000
101
10001
```

†La operación XOR $\oplus$ (OR exclusivo) de dos bits sigue la siguiente tabla:

x_1	x_2	$x_1 \oplus x_2$
0	0	0
0	1	1
1	0	1
1	1	0

Copia de seguridad corrupta

UNICOS

El servidor central de la OIE ha sufrido una leve corrupción de datos en su sistema de copias de seguridad. En uno de los sectores de memoria, existe una secuencia secreta a de $2 \cdot n+1$ enteros con valores entre 1 y n. Por diseño, cada valor debería estar respaldado apareciendo exactamente 2 veces, pero un *glitch* ha provocado que un único valor aparezca exactamente 3 veces.

Tu objetivo como auditor de sistemas es encontrar las 3 posiciones exactas de este dato corrupto.

Para no activar los protocolos de seguridad, no puedes descargar la secuencia completa. En su lugar, deberás utilizar una función de diagnóstico de la API realizando preguntas del siguiente estilo:

1. Elige un entero k y una secuencia de índices de tamaño k, $s_1, s_2, \ldots, s_k$, de valores **diferentes** entre 1 y $2 \cdot n + 1$.
2. Recibirás como respuesta el número de valores en $\{a_{s_1}, a_{s_2}, \ldots, a_{s_k}\}$ que aparecen **exactamente una vez**. A estos valores les llamamos **valores únicos**. Por ejemplo, si los valores $a_{s_1}, \ldots, a_{s_k}$ fueran $\{2, 1, 2, 3, 2, 3, 6, 7\}$ la respuesta sería **3**, ya que 1, 6 y 7 son los únicos valores que aparecen exactamente una vez. El 3 aparece dos veces y el 2 aparece tres veces, por lo que no se tienen en cuenta.

Interacción

Cada ejecución comenzará con un entero T, el número de casos a procesar. En cada caso se establecerá una comunicación con el juez según el siguiente protocolo:

- Primero debes leer un entero n, que indica que la secuencia oculta tiene $2 \cdot n + 1$ enteros.
- A continuación podrás realizar preguntas. Para preguntar por la secuencia de índices $s = \{s_1, s_2, \ldots, s_k\}$, debes imprimir `?` k s_1 s_2 … s_k, donde **los valores de s tienen que ser diferentes dos a dos**.
- Inmediatamente después de cada pregunta, recibirás un entero: la respuesta del juez a tu pregunta.
- Si lees `-1` después de una pregunta, indica que has superado el número máximo de preguntas o que tu pregunta es inválida. **Tu programa debe terminar inmediatamente** para obtener el veredicto “Respuesta incorrecta” en CMS. De otra forma, tu programa podría obtener veredictos arbitrarios.
- Cuando hayas resuelto el caso de prueba, deberás imprimir los 3 índices especiales (las 3 posiciones del dato corrupto) en cualquier orden utilizando el formato `!` i_1 i_2 i_3.
- Después de haber mostrado el resultado del caso, recibirás un `1` si tu respuesta es correcta y un `-1` si tu respuesta es incorrecta o inválida. Si recibes un `1`, puedes pasar a resolver el siguiente caso de prueba (si existe). Pero si recibes un `-1`, **tu programa debe terminar inmediatamente** para obtener el veredicto “Respuesta incorrecta” en CMS. De otra forma, tu programa podría obtener veredictos arbitrarios.

Recuerda que debes refrescar la salida cada vez que imprimas datos (usando `cout << endl` o `cout << flush` en C++, `System.out.flush()` en Java, `sys.stdout.flush()` en Python).

Para todos los casos de prueba de una misma ejecución (T), dispones de un límite global máximo de $100 \cdot T$ preguntas. Puedes exceder las 100 preguntas en un caso particular siempre y cuando lo compenses ahorrando preguntas en otros.

La secuencia secreta a se fija antes de empezar cada caso de prueba, es decir, el juez **no es adaptativo**.

Subtareas

1. (12 puntos) $T = 1$ y $n = 3$. Conseguirás los puntos si realizas no más de 100 preguntas.
2. (9 puntos) $T = 1$ y $n = 6$. Conseguirás los puntos si realizas no más de 100 preguntas.
3. (79 puntos) $T = 100$ y en todos los casos $n = 100$. Conseguirás los puntos en función de f, que depende del promedio q del número de preguntas sobre todos los casos, definida de la siguiente manera:

$$f(q) = \max(0, \min(79, 2 \cdot (63 - q)))$$

Para conseguir todos los puntos, el problema debe ser resuelto en como mucho $23.5 \cdot T$ preguntas.

Los casos de este problema han sido generados de manera aleatoria uniforme y para cada subtarea solo existe un único lote de casos.

Un ejemplo (ficticio) de caso de prueba con $T = 1, n = 2, a = [1, 2, 2, 1, 2]$ podría ser:

Juez	Concursante	Comentario
`1`		valor de T
`2`		valor de n
	`? 3 1 2 3`	
`1`		Entre los valores $a_1 = 1$, $a_2 = 2$ y $a_3 = 2$, solo hay un valor único: a_1.
	`? 1 4`	
`1`		
	`? 3 2 3 5`	
`0`		En $\{a_2, a_3, a_5\}$ no hay ningún valor único.
	`! 2 3 5`	Las posiciones de a con un valor repetido 3 veces son 2, 3 y 5
`1`		El caso se ha resuelto correctamente.

Ladrones en el museo

LADRONES

En el Colegio de San Gregorio se encuentra una de las sedes del Museo Nacional de Escultura, un museo con n salas conectadas entre sí por puertas. Desde cualquier sala se puede llegar a cualquier otra sala potencialmente teniendo que pasar por otras salas en el camino y, para evitar que los turistas se pierdan en el museo, este no tiene ciclos, es decir, tiene estructura de árbol.

Se sabe que hay varios ladrones en el museo y, usando el mínimo número de guardias posibles, se quiere arrestar a todos ellos.

Para arrestarlos se sigue la siguiente estrategia para explorar el museo:

- Todos los guardias empiezan en la sala 1, donde se encuentra la salida.
- Los guardias se van moviendo por diferentes salas coordinadamente hasta haber arrestado a todos los ladrones.

Pero claro, los ladrones también pueden moverse de sala en sala mientras no coincidan con un guardia. Para evitar que escapen, se requiere que toda sala que ha sido revisada y que está conectada a al menos una sala que no ha sido revisada contenga al menos un guardia.

¿Cuál es el mínimo número de guardias que hacen falta para realizar esta tarea si los guardias visitan las salas de forma óptima?

Entrada y salida

La primera línea contiene un entero T, el número de casos a procesar.

La primera línea de cada caso contiene n, el número de salas.

Las siguientes $n - 1$ líneas contienen dos enteros u, v, indicando que hay una puerta entre las salas u y v.

Debes imprimir un entero por caso, el mínimo número de guardias que hacen falta para realizar esta tarea.

Restricciones

- $1 \leq T \leq 10^5$
- $1 \leq n \leq 10^5$
- $1 \leq u, v \leq n$ y el grafo es un árbol (grafo conexo sin ciclos).
- La suma de n sobre todos los casos será como mucho 10^5.

Subtareas

1. (16 puntos) Todos los nodos del árbol tienen grado como mucho 2.
2. (11 puntos) El árbol es una estrella (existe un nodo tal que el resto de nodos están a distancia 1 de este).
3. (45 puntos) El árbol tiene como mucho 8 hojas, $T = 1$ y $n \leq 100$.
4. (28 puntos) Sin restricciones adicionales.

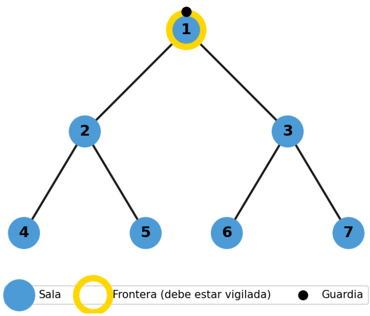
Ejemplo

Entrada
4 3 1 2 2 3 5 1 2 1 3 1 4 1 5 5 1 2 2 3 4 2 5 1 7 1 2 1 3 2 4 2 5 3 6 3 7

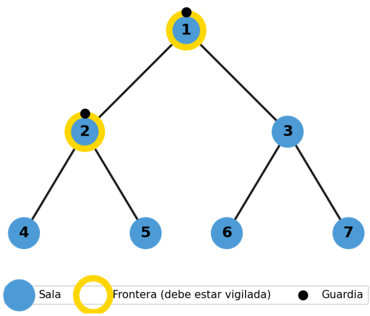
Salida
1 2 2 3

En el **cuarto caso** de ejemplo, el museo tiene forma de árbol binario con 7 salas. Para atrapar a los ladrones usando un mínimo de **3 guardias**, una estrategia óptima paso a paso es la siguiente:

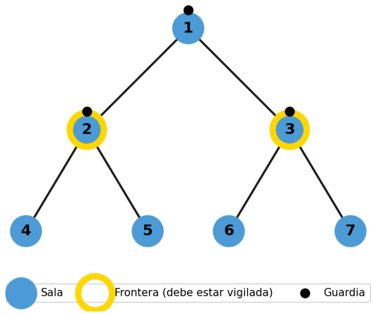
- **Paso 1:** Todos los guardias comienzan en la sala 1 (salida). Al estar conectada a salas sin revisar, actúa temporalmente como frontera.



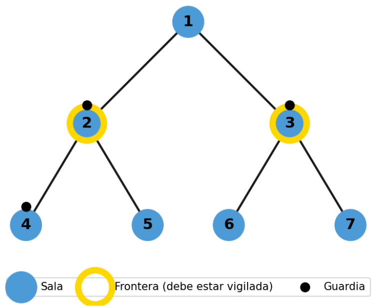
- **Paso 2:** Los guardias avanzan hacia la sala 2...



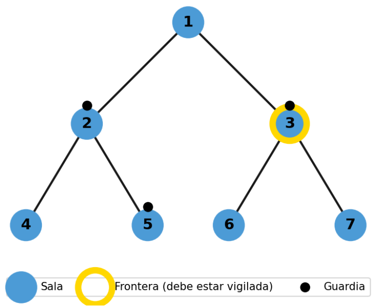
- **Paso 3:** ...y simultáneamente hacia la sala 3. Ahora las salas 2 y 3 son las nuevas fronteras que bloquean el escape de los ladrones. Notemos que en este paso solo se requieren realmente 2 guardias activos (uno en cada frontera), quedando el tercero a la espera.



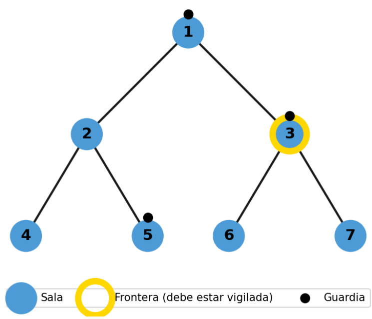
- **Paso 4:** Manteniendo las fronteras aseguradas en las salas 2 y 3, el tercer guardia se adentra para limpiar la sala 4. ¡Este es el momento crítico donde se necesitan los 3 guardias a la vez!



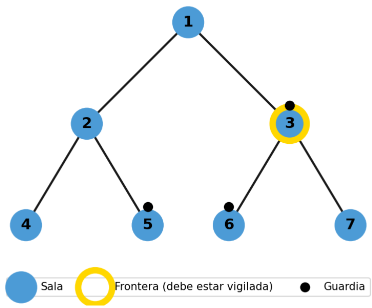
- **Paso 5:** El guardia termina de revisar la sala 4 y pasa a la sala 5. La rama izquierda queda asegurada.



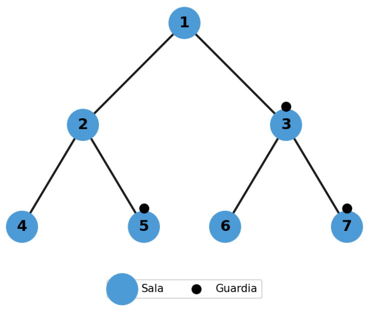
- **Paso 6:** Como la rama izquierda ya está limpia de ladrones, los guardias de esa zona pueden abandonar la frontera de la sala 2 y retroceder hacia la sala 1.



- **Paso 7**: Con la frontera mantenida firmemente en la sala 3, el guardia de apoyo avanza para limpiar la sala 6.



- **Paso 8**: Finalmente, el guardia revisa la sala 7. Todos los ladrones han sido acorralados y arrestados.



Camino a la OIE

AUTOBUSES

Ariadna y Darío viajan juntos para llegar a la estación de autobuses, desde donde cogerán un bus a Valladolid para asistir a la OIE 2026. Para llegar allí, han de atravesar la red de metro.

La red de metro consiste en n paradas, numeradas del 1 al n, y m líneas. La i-ésima línea tiene trenes unidireccionales que van de la parada a_i a la parada b_i (pero no al revés), y cuyo trayecto dura d_i minutos. Asumimos que los transbordos son inmediatos. Es posible que entre dos paradas haya más de una línea, y que haya paradas de las que no salen o entran líneas.

Cada parada es de un color: rojo o verde. Cuando Ariadna y Darío estén en una parada roja, Ariadna decidirá qué línea (de las que salen de esa parada) cogen. Cuando estén en una parada verde, Darío decidirá qué línea cogen. Si llegan a una parada de la cual no sale ninguna línea, entonces se tendrán que quedar allí para siempre, y jamás llegarán a la estación.

Ariadna y Darío tienen que ir de la parada 1, la de al lado de su casa, a la parada n, la estación de autobuses. Una vez lleguen a la estación de autobuses, no cogerán ningún metro más y su trayecto habrá terminado, independientemente del color de esta última parada.

A Ariadna le gusta llegar pronto a los sitios para ir con margen, por lo que intentará minimizar el tiempo total de viaje. Darío, que sabe que tiene trabajo esperándole, prefiere llegar lo más tarde posible, intentando maximizar el tiempo total.

Ambos tomarán decisiones de forma óptima conociendo la estrategia del otro. Ten en cuenta que para Darío, cualquier situación en la que nunca lleguen a la estación (un tiempo infinito) es siempre preferible a cualquier tiempo finito, mientras que Ariadna, por otro lado, preferirá cualquier tiempo finito a uno infinito.

¿Cuánto tardarán en llegar a la estación de autobuses?

Entrada y salida

La primera línea contiene un entero T, el número de casos.

Siguen T casos, cada uno con $m + 2$ líneas:

- La primera línea contiene n y m, el número de paradas y líneas de metro, respectivamente.
- La segunda línea contiene s, una cadena de n caracteres, donde el i-ésimo es ‘`R`’ o ‘`V`’ según si la i-ésima parada es roja o verde.
- Las siguientes m líneas contienen cada una 3 enteros, a_i b_i d_i, describiendo la i-ésima línea.

Para cada caso imprime una línea con, o bien un entero, el tiempo que tardarán en llegar, o bien la palabra “`INFINITO`” si no es posible llegar desde la parada 1 hasta la estación bajo ninguna circunstancia, o si Darío consigue forzar que se queden dando vueltas infinitamente sin llegar a la estación o atascados en una parada sin salida.

Restricciones

- $1 \leq T \leq 10^4$
- $2 \leq n \leq 2 \cdot 10^5$
- $1 \leq m \leq 2 \cdot 10^5$

- $1 \leq d_i \leq 1000$
- a_i, b_i son enteros diferentes entre 1 y n.
- La suma de n sobre todos los casos es como mucho $2 \cdot 10^5$.
- La suma de m sobre todos los casos es como mucho $2 \cdot 10^5$.

Subtareas

1. (13 puntos) Todas las paradas son rojas.
2. (23 puntos) Todas las paradas son verdes.
3. (15 puntos) $a_i < b_i$ para todo $1 \leq i \leq m$.
4. (24 puntos) Tanto la suma de n como la suma de m para todos los casos son como mucho 2000.
5. (25 puntos) Sin restricciones adicionales.

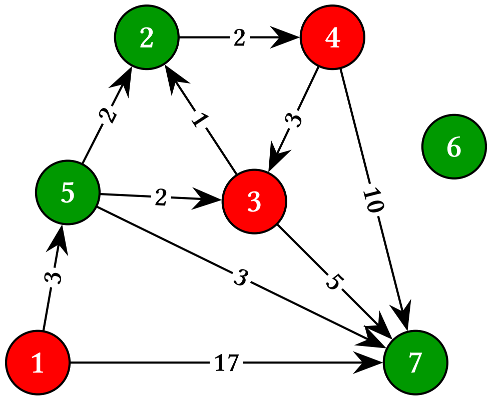
Ejemplo

Entrada

```
3
7 10
RVRRVVV
1 5 3
1 7 17
5 7 3
5 2 2
5 3 2
2 4 2
3 2 1
3 7 5
4 7 10
4 3 3
3 3
VRR
1 2 7
1 2 10
1 3 10
4 6
VRVR
1 2 1
1 4 1
3 2 1
3 4 1
2 1 5
2 3 5
```

Salida

```
15
INFINITO
INFINITO
```



En el **primer caso**, empieza eligiendo Ariadna desde la parada 1 (ya que es roja), que debe elegir entre ir a la parada 5, tardando 3 minutos, o directamente a la 7 (la estación de autobuses) y acabar, pero tardaría 17 minutos. Elige ir a la 5, pues así acabarán llegando más rápido y ella quiere llegar pronto.

Ahora elige Darío desde la parada 5 (ya que es verde), que debe elegir entre ir a la 7 en 3 minutos, a la 3 en 2 minutos o a la 2 en 2 minutos. Como quiere llegar lo más tarde posible, no le interesa la primera opción. Si eligiera ir a la 3, Ariadna iría desde allí directa a la 7 y la duración total del viaje acabaría siendo de $3 + 2 + 5 = 10$ minutos. Como yendo a la 2 puede acabar consiguiendo un viaje más largo, elige ir a la 2.

Ahora a Darío le toca elegir en la parada 2 (ya que es verde), pero solo tiene una opción: ir a la parada 4 en 2 minutos, así que está obligado a elegir esa línea.

Ahora Ariadna elige desde la parada 4. Sus opciones son ir a la 7 directamente, tardando 10 minutos, o ir primero a la 3. Como ve que desde la 3 puede ir en 5 minutos a la 7, llegan antes si elige ir a la 3 primero.

Finalmente Ariadna elige desde la parada 3. Puede o bien ir a la 2 en 1 minuto o bien a la 7 (la estación de autobuses) en 5 minutos, y claramente prefiere la segunda.

Finalmente llegan a la estación de autobuses, tras un trayecto que dura un total de $3 + 2 + 2 + 3 + 5 = 15$ minutos.

Los otros dos casos son las dos maneras en que Darío puede conseguir que nunca lleguen a la estación.

En el **segundo caso**, si Darío elige ir a la estación 2 (por cualquiera de las dos líneas que van), entonces llegan a una estación sin líneas, y por tanto nunca pueden salir de ahí.

En el **tercer caso**, si Darío siempre elige ir a la estación 2, puede conseguir que se queden dando vueltas infinitamente, sin llegar jamás a la 4.