Saltos

Vueltas es un problema cuya solución óptima requiere de cierto análisis matemático. Las matemáticas que se requieren son muy básicas, aritmética y nociones de teoría de números. El problema buscaba evaluar que tanta capacidad tenía el alumno de analizar matemáticamente el planteamiento de un problema.

La idea más simple para resolver el problema es guardar en memoria el tablero, ir marcando las posiciones por las que ya se pasó, simular los saltos y cuando se regrese a una posición previamente visitada, dar como respuesta el número de saltos que se han dado. Sin embargo el tablero es muy grande (10millones x 10millones) y almacenarlo en memoria para ir marcando las casillas visitadas requiere de 100TB de RAM.

El alumno debía ser capaz de darse cuenta de que la memoria requerida era excesiva y buscar una forma alterna para saber si había caído en una casilla previamente visitada.

En muchos problemas, no sólo de informática, sino de la vida. El camino más corto hacia la respuesta es plantearse la pregunta correcta. Las preguntas que debían haberse formulado en la mente del alumno debían ser las siguientes (en el orden que las escribo):

- ¿Es cierto que en algún salto voy a regresar a una casilla visitada? La respuesta afirmativa a esta pregunta se obtiene dándose cuenta de que el número de casillas en el tablero es finita, sin embargo el número de saltos que se pueden dar no, eventualmente el número de saltos excederá el número de casillas en el tablero, por lo que forzosamente se tiene que volver a caer en una casilla que ya se visitó previamente.
- OK, ya vi que en algún momento visitaré de nuevo una casilla. ¿Puedo visitar cualquier casilla de las que ya visité? La motivante de esta pregunta debe ser el hecho de darse cuenta de que para saber si ya visitaste una casilla tienes que de alguna forma mantenerla en memoria, ya que el número de casillas es, como dijimos, 100 millones de millones, esto no es posible. Antes de ponerse a almacenar, vale la pena preguntarse si en efecto es necesario recordar todas las casillas por las que se ha pasado.
    - Dado que el salto está definido, estando en cualquier casilla del tablero, sólo se puede pasar a otra casilla específica, es decir si estás en la casilla $\boldsymbol{a_0}$ y saltas, forzosamente caerás en la casilla $\boldsymbol{a_1}$**,** no hay posibilidad de cambiar la casilla en donde vas a caer ya que el salto siempre es el mismo. Más aún, si estas en la casilla $\boldsymbol{a_1}$ el único lugar de donde podías venir era de $\boldsymbol{a_0}$**.**
    - De lo anterior puede verse que si en algún momento caes en una casilla que y habías visitado, entonces hay dos opciones:
        - La casilla de donde venías también ya había sido visitada, o
        - La casilla en la que caíste es la casilla de donde empezaste.
    - Por lo tanto, la primera casilla que repetirás será siempre la primera casilla. No es necesario almacenar todas las casillas por las que has pasado, basta únicamente con fijarse si vuelves a la casilla original.

Solución parcial: 50 puntos

La solución parcial utiliza lo deducido en los puntos anteriores para ir simulando los saltos y checando si volvió a la casilla original. La razón por la que esta solución no obtiene el 100% de los puntos es porque hay tableros en los que el número de saltos es de muchos millones y la simulación tarda más tiempo del permitido.

Solución óptima: 100puntos

Un buen hábito cuando se tiene una solución es preguntarse cómo se va a desempeñar dicha solución en casos extremos. Haciendo este ejercicio es fácil encontrar ejemplos en los que el número de saltos a dar es de varios millones. Un segundo no alcanza para ejecutar muchos millones de sumas, de modo que se debe buscar una solución que haga menos cálculos.

Dado que simular es muy tardado, debe existir una forma de calcular de manera fácil de calcular en cuantos saltos se vuelve al inicio. Como siempre es necesario ir analizando las propiedades del problema.

- La primera propiedad a notar es que el salto está dividido en dos partes, la vertical y la horizontal y ambas son independientes una de la otra. Si es posible saber cada cuando el salto vertical nos lleva de regreso a la fila de origen y cada cuando el salto horizontal nos lleva a la columna de origen. Entonces sabemos que cada múltiplo de estos dos números nos llevara a la fila y columna de origen. Por lo tanto teniendo esos números basta con buscar el mínimo común múltiplo de ambos para tener la solución.

La pregunta entonces es: ¿Cada cuánto, tomando por ejemplo el salto horizontal, volvemos a la columna de origen?

- Para volver a caer en la columna de origen (pensando en los saltos horizontales), necesitamos que los saltos nos desplacen un número de columnas que sea un múltiplo de la longitud del tablero, es decir, si el tablero tuviera 1000 columnas de ancho, entonces volveremos a la columna original si avanzamos 1000 columnas, 2000 columnas, 3000 o cualquier otro múltiplo del ancho del tablero.
- Lo que hay que buscar entonces es el mínimo común múltiplo entre ambos números, la longitud del tablero y la longitud del salto y dividirlo entre la longitud del salto.

Hasta aquí ya sabemos en cuantos saltos se vuelve a la columna original y a la fila original, también sabemos que el mínimo común múltiplo de ambos será la cantidad de saltos necesaria para volver a la casilla original. El único punto que queda pendiente es ¿Cómo se calcula el mínimo común múltiplo de dos números?

Una forma es de nuevo ir incrementado cada número por sí mismo (a manera de irlo multiplicando por 1, por 2, por 3, etc.) hasta que ambos lleguen al mismo número. Ese será el mínimo común múltiplo de ambos. Aunque este algoritmo es correcto, de nuevo, puede llegar a ser lento, ya que puede implicar un gran número de sumas antes de encontrarse.

Si recordamos nuestra definición de primaria de un mínimo común múltiplo sabremos que se encuentra al factorizar en primos ambos números, tomar la intersección de primos entre ambos y multiplicar, la intersección por los primos no comunes en ambos números. Factorizar en primos es otra cosa que puede tomar mucho tiempo, para empezar, porque necesitamos tener una lista de todos los primos a probar. Sin embargo, es útil notar que la intersección de primos entre ambos números es además lo que se conoce como máximo común divisor de ambos. Si podemos calcular la intersección de primos de ambos números, podemos obtener los primos no comunes simplemente dividiendo el número original entre el máximo común divisor. En otras palabras:

Mínimo común múltiplo(a,b)=a * b / máximo común divisor(a,b)

Obtener el máximo común divisor de ambos números es algo computacionalmente más sencillo. La forma más común es utilizar el algoritmo de Euclides, conocido desde hace más de 2000 años. El algoritmo se basa en la siguiente característica:

- Sean a y b dos números enteros donde $a < b$. Al dividir b/a se obtiene un cociente q y un residuo r.
- El fundamento del algoritmo de Euclides se basa en que el máximo común divisor de a y b es el mismo que el máximo común divisor entre a y r.

Demostrar que el fundamento del algoritmo de Euclides es cierto es relativamente sencillo y para no extenderme demasiado en este problema queda como tarea para el lector. Como pista sugiero un método visual (con varitas) donde el problema se traduce en ver ¿Cuál es la varita más grande que cabe de manera exacta dentro de las otras dos varitas? Y tratar de ver por qué es necesario que el mínimo común divisor de a y de b divida también de manera exacta a r.

Como última nota para este problema, es necesario observar que en algunas multiplicaciones un entero de 32 bits puede desbordarse, de modo que es necesario utilizar un tipo de datos de 64 bits para asegurar que el problema se resuelve correctamente en el 100% de los casos.