

Vueltas

Este es el problema más sencillo del día 2, y fue en el que más alumnos obtuvieron puntos. Como se esperaba, hubo una buena cantidad de alumnos con 100 puntos y una cantidad similar de alumnos con 50 puntos (eran los obtenidos por la solución parcial). Fueron muy pocos los alumnos que obtuvieron un puntaje distinto de 100, 50 ó 0 y en esos casos se debió principalmente a errores al momento de implementar.

El problema pide que se volteé un tablero que puede ser de tamaño máximo 1000 x 1000 al cual se le puede dar un máximo de 50,000 vueltas.

Solución parcial: 50 puntos

El tablero es representable en memoria, ya que 1000x1000 son 1,000,000 de enteros los cuales, tomando enteros de 4 bytes representan un gasto total de 4MB de memoria. El límite para el problema era de 64MB, por lo que estamos muy por abajo.

La solución parcial es la que representa el tablero en memoria y va volteando el tablero conforme se le indica en el archivo de entrada. El problema con esta solución es que puede llegar a ser muy lenta. Analizando la complejidad de la misma podemos ver que en cada vuelta del tablero volteamos todas las posiciones del tablero, es decir, potencialmente 1,000,000 de asignaciones a memoria. Si repetimos esto 50,000 tenemos que nuestro programa puede realizar potencialmente hasta 50,000,000,000 de operaciones. Suponiendo que una máquina moderna puede realizar algunas decenas de millones de operaciones en un segundo, esta solución podría tardar del orden de 500,000 segundos en terminar (¡Alrededor de una semana!)

Sin embargo, en la mitad de los casos el número de vueltas no era tal que una implementación que fuera simulando todas las vueltas corre en tiempo.

Solución óptima: 100 puntos

Para la solución óptima, el alumno tiene que darse cuenta de dos cosas:

- Dar dos vueltas seguidas en un sentido (vertical u horizontal) es lo mismo que no haber dado ninguna vuelta. Esto es algo sencillo de ver, el alumno que lo note podrá darse cuenta de que en realidad sólo importa si el número de vueltas en un sentido es par (indica que no hay que dar ninguna vuelta) o impar (indica que basta dar una vuelta para llegar al estado final).
- El orden en de las vueltas es conmutativo: La primera propiedad es sencilla de ver, sin embargo convencerse de que no importa si en medio de vueltas verticales metes una horizontal por ejemplo, es algo un poco menos obvio. Menos aún el hecho de que puedes tomar todas las órdenes de vueltas, ordenarlas en el orden que quieras, aplicarlas y al final el resultado será el mismo. Una forma de verlo es darse cuenta es observar que las vueltas verticales afectan la fila final de un número, pero nunca su columna, mientras que las horizontales afectan sólo la columna, pero no la fila. Experimentando un poco, en un tiempo muy corto puede verse que el orden de las vueltas no tiene efecto sobre el resultado final.

Por tanto la solución óptima lee el tablero en memoria, cuenta el número de vueltas verticales y horizontales, si el número de vueltas verticales es impar, aplica sólo una vuelta vertical. Si el número de vueltas horizontales es impar aplica sólo una vuelta horizontal. Posteriormente escribe el estado final de la tabla.