Tercera parte: Cambios

Estrategia para demostrar que el problema está en NP

Se plantea el problema como:

Dado un tablero $n \times m$, una lista de k barcos y las restricciones de filas y columnas, queremos saber si existe una disposición válida de barcos que cumpla con todas las restricciones.

Solución candidata

Construir un algoritmo que, dado un tablero candidato, verifique en tiempo polinomial si:

- Los barcos están correctamente colocados.
- Se respetan las restricciones de las filas y columnas.
- Se respetan las restricciones de adyacencia.

Algoritmo de verificación

Dado un tablero $n \times m$ con una configuración candidata, se plantea un algoritmo que realiza la verificación en pasos. El algoritmo propuesto verifica lo siguiente:

1. Verificar restricciones de filas

Para cada fila n del tablero:

- Contar cuántas casillas están ocupadas.
- Verificar que el conteo sea igual a la restricción dada para esa fila.

Complejidad: $O(n \cdot m)$.

2. Verificar restricciones de columnas

Para cada columna m del tablero:

- Contar cuántas casillas están ocupadas.
- Verificar que el conteo sea igual a la restricción dada para esa columna.

Complejidad: $O(n \cdot m)$.

3. Verificar restricciones de adyacencia

Para cada casilla ocupada en el tablero:

- Verificar que no haya casillas ocupadas en las posiciones adyacentes (arriba, abajo, izquierda, derecha, y las diagonales).

Complejidad: $O(n \cdot m)$, porque para cada casilla verifica a lo sumo 8 sus casillas adyacentes.

Posible cuarta, verificar los barcos colocados

- Contar que todos los barcos de la lista hayan sido colocados una sola vez y que se usen todos.

Complejidad: $O(k)$ siendo k la cantidad de barcos.

Detalle de la complejidad

En el peor de los casos, si todas estas comprobaciones se realizan por separado (que no sería lo óptimo):

- Verificación de las restricciones de filas: $O(n \cdot m)$.
- Verificación de las restricciones de columnas: $O(n \cdot m)$.
- Verificación de las restricciones de adyacencia: $O(n \cdot m)$.
- Verificación de cant. de barcos colocados: $O(k)$.

El costo total del algoritmo sería:

$$O(n \cdot m) + O(n \cdot m) + O(n \cdot m) + O(k) = 3 \cdot O(n \cdot m) + O(k) = O(n \cdot m)$$

Conclusión

El algoritmo descrito verifica si una solución candidata es válida en tiempo $O(n \cdot m)$, que es polinomial con respecto al tamaño del tablero y las restricciones. Esto demuestra que el problema está en NP, porque verificar una solución candidata puede hacerse en tiempo polinomial.