

# **PRÁCTICA 7 ALGORÍTMIA**

Nahiara Sánchez García – UO295645

## INTRODUCCIÓN:

Esta práctica está basada en la resolución del problema del cuadrado numérico mediante algoritmos de ramifica y poda. Este tipo de algoritmos son una variante de vuelta atrás (backtracking) mejorado en algunas situaciones. Básicamente construye un árbol de anchura máxima, podando aquellas ramas en las que las soluciones ya no están siendo óptimas. La solución se encuentra cuando todos los nodos del árbol están podados o resueltos. Es importante tener en cuenta que durante esta práctica solo se realiza el algoritmo para hallar una solución, pues si hubiera que hallar todas posiblemente fuera más eficiente mediante algoritmos de vuelta atrás.

## OBJETIVO:

El principal objetivo es implementar un algoritmo con la técnica de ramifica y poda para resolver el problema de manera óptima. Además, es necesario encontrar un heurístico que, en la medida de lo posible, mejore los tiempos obtenidos con backtracking.

1. Tras implementar la clase CuadradoNumericoRamifica en java que calcule, para una entrada de tablero válida, una solución de la forma más eficiente posible, se obtienen los siguientes tiempos:

Caso de prueba:	Tiempo para la primera solución (vuelta atrás) - ms	Tiempo para la primera solución (ramifica y poda) - ms
Test00	FdF	71
Test01	FdF	287
Test02	FdF	FdF
Test03	625	3982
Test04	7113	21252
Test05	490	19880
Test06	79	824
Test07	17368	128

Cabe recalcar que el test 2 no funciona correctamente, pues no consigue encontrar una solución. Además, los test 4 y 5 producen un error distinto: Java heap space. Esto quiere decir que la Máquina Virtual de Java se quedó sin memoria en el heap disponible para asignar objetos nuevos, es decir, el algoritmo ocupa demasiada cantidad de memoria.

2. A continuación se procederá a explicar cómo se intentó calcular el heurístico. El método encargado de calcular el valor del heurístico recorre todo el tablero y cuenta el número de casillas que contienen un "?". Si se comprueba una columna específica y es incorrecta, el heurístico toma el valor del máximo entero. Lo mismo se hace en el caso de las filas. Si el valor heurístico final calculado es distinto al máximo entero, entonces el valor del heurístico será correcto.
3. En este caso particular, se han obtenido mejores tiempos para backtracking que para ramifica y poda. Esto puede suceder por diversos motivos, como que el algoritmo de ramifica y poda no esté correctamente optimizado; una heurística poco efectiva, pues es muy

importante para la poda de ramas... Posiblemente esté mejor implementado backtracking que ramifica y poda.