

Tema 2

Opcional (2 EC): Demostrar proposición 2.1 y 2.2

Opcional (1 EC): Demostrar que multiplicar el exponente por un factor altera el coste computacional (diapositiva 25 Tema2.pdf)

Opcional (1 EC): Demostrar que si $T(n) = \max\{f(n), g(n)\}$ entonces $T(n) = \Theta(f(n) + g(n))$ (diapositiva 26 Tema2.pdf)

Calcular el coste (número de primitivas) del peor caso y del mejor caso del algoritmo de ordenación por Selección.

Calcular el coste (número de primitivas) del peor caso y del mejor caso del algoritmo de ordenación por Inserción.

Implementar una función que calcula el factorial de un número, y determinar su coste asintótico (Big-O, Big-Omega y Big-Theta si existe).

Implementar una función que recibe dos matrices cuadradas de $N \times N$ y devuelve la matriz resultante de la multiplicación. Determinar su coste asintótico.

Implementar una función que recibe una matriz cuadrada de $N \times N$ y devuelve el determinante de dicha matriz desarrollando siempre por la primera fila. Determinar su coste asintótico.

Implementar una función que recibe una matriz cuadrada de $N \times N$ y devuelve la traza de dicha matriz. Determinar su coste asintótico.

Opcional (2 EC): Demostrar las propiedades de Reflexividad, Transitiva y Simetría de Θ

Opcional (2 EC): Demostrar las propiedades de Reflexividad, Transitiva de O . Poner un ejemplo donde no se cumple la propiedad Simetría.