

### Guía Práctica No. 4: Decrease & Conquer

Se sugiere que la resolución de esta práctica se complete antes del **12 de Abril de 2019**.

1. Implemente en Java soluciones Decrease & Conquer con decremento por una constante y por un factor constante para el problema de multiplicar dos números enteros. Calcule además el número de sumas efectuadas por su algoritmo, en función del tamaño de la entrada.  
En relación a una de sus soluciones, considere que es más económico computar  $n + n$  o  $2 \times n$ ? Justifique su respuesta.
2. Implemente una solución para el problema de la moneda falsa que, dadas  $n$  monedas ( $n > 1$ ) aparentemente iguales, se desea encontrar la moneda falsa, que es más liviana que las originales. Utilice una balanza clásica de platillo (la misma del signo de libra) para determinar el peso.
3. Utilizando la técnica Decrease & Conquer, diseñe un algoritmo para encontrar los elementos mayor y menor de una secuencia de  $n$  enteros positivos. Implemente su algoritmo en Haskell.
4. Utilizando la técnica Decrease & Conquer, diseñe un algoritmo para encontrar el índice del mayor elemento de una secuencia de  $n$  enteros positivos. Piense en una alternativa a este algoritmo diseñado utilizando Fuerza Bruta, y compare implementaciones para estos dos algoritmos en Haskell.  
Realice además el análisis correspondiente para calcular cuántas comparaciones de elementos son realizadas por ambos algoritmos en el peor caso.
5. Diseñe usando Decrease & Conquer e implemente en Java o Haskell un programa que, dada una lista de asignaturas de un plan de estudios y un conjunto de correlatividades de cursado (cada una de las cuales indica un pre-requisito para el cursado de una materia), determine si el plan es *consistente*, es decir, si el mismo contiene correlatividades cíclicas. Su algoritmo no sólo debe emitir el *veredicto* (consistente o inconsistente), sino que en caso de consistencia debe dar un ejemplo, es decir una forma de cursar asignaturas en cuatrimestres consecutivos de manera tal de respetar las correlatividades.
6. Diseñe usando Decrease & Conquer e implemente en Java y Haskell un programa que, dadas dos secuencias de caracteres, construya la subsecuencia común a ambas de longitud máxima. Se entiende por *subsecuencia* una cadena de caracteres que se deriva de la original mediante la eliminación de caracteres pero sin cambiar el orden de los caracteres en la secuencia original.