

Análisis de Algoritmos (I)

Profesor: Carlos Zerón Martínez
Ayudante: Edwin Antonio Galván Gamez

Reposición de Tarea 4: Diseño de Algoritmos
Fecha de entrega: Jueves 6 de Junio del 2019, 18 hrs

1. Una subsecuencia de una secuencia de números reales S decimos que es consecutiva si contiene elementos que están en posiciones consecutivas de S . Por ejemplo, para $S = \{5, 15, -30, 10, -5, 40, 10\}$, $S_1 = \{15, -30, 10, -5\}$ es una subsecuencia consecutiva de S y $S_2 = \{-30, -5, 40\}$ también es una subsecuencia de S pero no es consecutiva. Decimos que S' es una subsecuencia consecutiva máxima de una secuencia S si la suma de los elementos que la conforman es la máxima para cualquier subsecuencia consecutiva de S . La subsecuencia vacía (con cero elementos) es una subsecuencia consecutiva cuya suma es definida como cero.

Diseña un algoritmo, usando programación dinámica, que encuentre los elementos de la subsecuencia consecutiva máxima de una secuencia S de n números reales. Describe en pseudocódigo el algoritmo con sumo detalle, demuestra su integridad y analiza su complejidad. (*Hint: Para cada $j = 1, 2, \dots, n$ considera subsecuencias consecutivas que terminan exactamente en la posición j -ésima. Puedes empezar por calcular la suma de la subsecuencia consecutiva máxima y luego recuperar sus elementos.*)

2. **Ejercicio teórico - práctico.** Considera el siguiente problema: Dado un arreglo $A[a..b]$ de n números reales, donde $n = b - a + 1$ y otro número real x , determinar si x está en $A[a..b]$.
 - a) Diseña un algoritmo que resuelva el problema usando el paradigma *Divide y Vencerás*. (*Hint: intenta usar una estrategia parecida a la búsqueda binaria*) usando inducción matemática.
 - b) Define precondiciones y poscondiciones de tu algoritmo y descríbelo usando pseudocódigo.
 - c) Analiza la complejidad de tu algoritmo. ¿Es éste más eficiente que la búsqueda lineal?
 - d) Implementa tu algoritmo en Java.

Suerte!