

Análisis de Algoritmos (I)

Profesor: Carlos Zerón Martínez

Ayudantes: José Antonio Vilchis Salazar, Yessica Janeth Pablo Martínez

Programa 3: Diseño de Algoritmos

Fecha de entrega: Viernes 29 de Abril del 2022.

Considera el problema de la subsecuencia aumentante de máxima longitud *Longest Increasing Subsequence Problem*: Dada una lista $S = [s_1, s_2, \dots, s_n]$ de $n \geq 1$ números distintos, una subsecuencia creciente de S es un subconjunto de números $T = [t_1, t_2, \dots, t_m]$ de S tomados en el orden en que aparecen dentro de S y que definen un orden creciente, esto es, $\forall i, j$ tales que $1 \leq i < j \leq m : t_i < t_j$. El objetivo consiste en encontrar la subsecuencia aumentante con mayor número de elementos de S .

Por ejemplo, consideremos la secuencia $S = \{2, 4, 3, 5, 1, 7, 6, 9, 8\}$. La longitud de las soluciones óptimas es 5 y una de las 8 posibles soluciones es $\{2, 3, 5, 6, 8\}$.

Implementa el algoritmo que se subió a Classroom, dedicado a calcular los valores de dos arreglos, el que contiene las longitudes de las subsecuencias y el que refleja los elementos que preceden a los que terminan en cada uno de los elementos de la secuencia original, utilizando programación dinámica; debes incluir también el algoritmo que permite recuperar una subsecuencia que resuelva el problema para un ejemplar dado, siguiendo la idea descrita en el texto. El programa también deberá permitir el ingreso de los elementos de la secuencia que conforma el ejemplar, ya sea por archivo o por medio de la terminal, y deberá presentar una subsecuencia de máxima longitud, por la misma terminal o representada en un archivo.

Suerte!