

# Estructuras de Datos 2021-2

Facultad de Ciencias, UNAM

## Práctica 1

Autor: Leonardo Gallo Guerrero

Revisión: Virginia Teodosio Procopio , Alain Acevedo Mejía

Entrega : 11:59 am del 12 de julio del 2021

### Objetivos

Que el alumno se introduzca en el uso del concepto de tiempo de ejecución de un programa para poder determinar cómo afecta en el desempeño de su ejecución, y que, además, pueda identificar qué otros factores afectan su eficiencia.

### Descripción

En esta práctica se compara la funcionalidad y la complejidad de cuatro algoritmos, cada uno de los cuales resuelve el problema de subsecuencia de suma máxima.

*Subsecuencia de suma máxima:* Sea  $Q = a_1, a_2, \dots, a_n$  una secuencia de  $n$  enteros (posiblemente negativos), el objetivo es encontrar naturales  $i, j$  (con  $1 \leq i \leq j \leq n$ ), representando una subsecuencia contigua  $a_i, \dots, a_j \in Q$ , tales que  $\sum_{k=i}^j a_k$  sea la mayor para cualesquiera  $i, j$ . Si para todo  $a \in Q$ ,  $a < 0$ , entonces la suma máxima es 0.

### Ejercicios

Usando el código que se proporciona hacer las siguientes modificaciones y pruebas.

1. [2.5 puntos] Tomar el tiempo de ejecución para cada algoritmo usando secuencias de enteros muy grandes (100, 200, 300, 400, 500 elementos). El tiempo de ejecución se puede calcular con el reloj de java (se puede usar el método `System.currentTimeMillis()`).
2. [2.5 puntos] Responder las siguientes preguntas:
  - i) ¿Siempre se obtienen los mismos resultados de tiempo al usar las mismas secuencias de enteros?
  - ii) ¿Qué factores alteran los tiempos de ejecución?
  - iii) ¿Qué puede minimizar estos factores?
  - iv) ¿Cuál implementación esperas que tenga mejor tiempo de ejecución?  
¿Por qué?
3. [2.5 puntos] Graficar y comparar los resultados obtenidos en los puntos 1 y 2.
4. [2.5 puntos] Elabora un análisis, no necesariamente formal, de lo que tarda cada algoritmo en ejecutarse respecto a una entrada de  $n$  elementos, basándose sólo en el algoritmo y no en los factores de su implementación.

*Sugerencia:* Para crear las secuencias de enteros para probar los ejercicios 1 y 2, puedes obtener números aleatorios con los métodos random disponibles en la bibliotecas de java y generar arreglos de enteros.

### Entrega

Revisar el documento con las especificaciones de la entrega de las prácticas ubicado en el almacén del curso en GitHub.