



UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingeniería Informática y Matemáticas

PRÁCTICA 1: ANÁLISIS DE EFICIENCIA DE ALGORITMOS

Doble Grado Ingeniería Informática y Matemáticas

Autores:
Moya Martín Castaño
Javier Gómez López
Jose Alberto Hoces Castro

Marzo 2022



Este trabajo se distribuye bajo una licencia CC BY-NC-SA 4.0.

Eres libre de distribuir y adaptar el material siempre que reconozcas a los autores originales del documento, no lo utilices para fines comerciales y lo distribuyas bajo la misma licencia.

creativecommons.org/licenses/by-nc-sa/4.0/

Índice

1. Introducción	3
1.1. Análisis de la eficiencia teórica	3
1.2. Análisis de la eficiencia empírica	3

1. Introducción

Esta primera práctica, **Práctica 1**, consiste en el análisis de eficiencia de algoritmos, consiste en tres partes distintas:

- **Análisis de la eficiencia teórica:** estudio de la complejidad teórica del algoritmos (Mejor caso, peor caso y caso promedio).
- **Análisis de la eficiencia empírica:** ejecución y medición de tiempos de ejecución de los algoritmos estudiados.
- **Análisis de la eficiencia híbrida:** obtención de las constantes ocultas

A continuación, se explican en más profundidad dichas partes.

1.1. Análisis de la eficiencia teórica

El análisis de la **eficiencia teórica** consiste en analizar el tiempo de ejecución de los algoritmos dados para encontrar el peor de los casos, es decir, en qué clase de funciones en notación \mathcal{O} grande se encuentran. Para ello, hemos utilizado las técnicas de análisis de algoritmos vistas en clase y en la asignatura *Estructura de Computadores*.

1.2. Análisis de la eficiencia empírica

Para el análisis de la **eficiencia empírica**, hemos ejecutado los algoritmos en cada uno de nuestros equipos bajo las mismas normas y condiciones, hemos medido el tiempo de ejecución de dichos algoritmos con la biblioteca `<chrono>`, basándonos en la siguiente estructura del código:

Meter arriba texto de introducción. ¿METEMOS INDICE?

Para el análisis de la eficiencia en distintos aspectos a tener en cuenta hemos realizado las siguientes tareas: -

Comparación de tiempos entre ejecuciones en distintos ordenadores.

- Comparación de tiempos entre ejecuciones en distintos sistemas operativos.

- Comparación de tiempos entre ejecuciones para el peor y el mejor caso en inserción y selección.

- Comparación de tiempos entre ejecuciones utilizando optimización y sin utilizarla.

Para los algoritmos de inserción y selección hemos visto que la eficiencia híbrida nos aporta las siguientes funciones: (ESCRIBIR AQUÍ LAS FUNCIONES QUE LE HAN SALIDO A JOSÉ ALBERTO)

Inserción: $f(x) = 1,04896 * 10^{-9} * x^2 + 2,1386 * 10^{-7} * x - 0,0120955$

Selección: $f(x) = 1,29484 * 10^{-9} * x^2 - 7,43377 * 10^{-6} * x + 0,0733569$

(ESCRIBIR AQUÍ LOS TIEMPOS QUE LE HAN SALIDO A JOSÉ ALBERTO)

Vemos que para estas funciones, la ejecución de un tamaño de vector de 500000 datos, se tardaría 323.8048345 segundos para una única ejecución de tamaño 500000. Esto multiplicado a 15 ejecuciones para obtener el promedio nos daría que para obtener el tiempo tamaño de 500000 se tardan $48570,72518 \simeq 1,3491$ horas. El mismo razonamiento lo podemos aplicar para obtener el tiempo de ejecución para tamaño 500000 con selección, donde para una única ejecución sale un tiempo de 320.0664719 segundos. Multiplicando este valor ahora por 15 tenemos que selección tarda $4800,997079 \simeq 1,3336103$ horas. Teniendo esto en cuenta hemos decidido llegar hasta un tamaño de vector de 200000 datos.