



# Estructuras de Datos y Algoritmos (EDA)

## Tarea 1: Análisis de Algoritmos y Tiempo de Procesamiento

Prof: José M. Saavedra Rondo  
Ayudantes: Lukas Pavez & Vicente Romero

**Fecha de entrega:** 25 de agosto de 2024

### 1. Objetivo

Comprender la importancia del diseño eficiente de algoritmos a través de una tarea experimental sobre análisis de tiempos de ejecución de algoritmos de búsqueda y ordenación.

### 2. Descripción

Dado un archivo de  $N_D$  palabras en inglés, que representa un diccionario, y un conjunto de  $N_q$  términos, que representan la consultas, se pide implementar dos tipos de algoritmos de búsqueda para determinar qué palabras existen en el diccionario. Los algoritmos a evaluar son:

1. **Algoritmo Secuencial (AS):** este algoritmo busca cada término en secuencia dentro de los  $N_D$  elementos del diccionario.
2. **Algoritmo Búsqueda Binaria (ABB):** este algoritmo requiere tener un diccionario ordenado para luego aplicar la estrategia de búsqueda binaria produciendo un tiempo de orden  $O(\log(n))$ . Para este fin, deberás elegir un algoritmo eficiente de ordenación.

La Figura 1 ilustra el esquema general de la tarea. Aquí, el algoritmo de búsqueda, luego de analizar cada consulta deberá retornar el número de términos encontrados en el diccionario así como el tiempo promedio de búsqueda por cada término, en milisegundos.

#### 2.1. Experimentación

1. El número de consultas  $N_q$  es fijo y es igual a 1MM.
2. El tamaño del diccionario varía según  $N_D \in \{10K, 50K, 100K, 200K, 400K\}$
3. Para cada uno de los diccionarios de tamaño  $N_D$ , deberás estimar el tiempo de búsqueda promedio de cada término de la consultas mostrando una tabla como el del Cuadro 1.

En la tabla indicada, **AS** es el tiempo promedio de búsqueda del algoritmo secuencial. **ABB** es tiempo promedio de búsqueda sin considerar el tiempo de ordenación (pero debes ordenar). Finalmente **ABB + Sort** indica el tiempo promedio de ABB considerando el tiempo de ordenación.

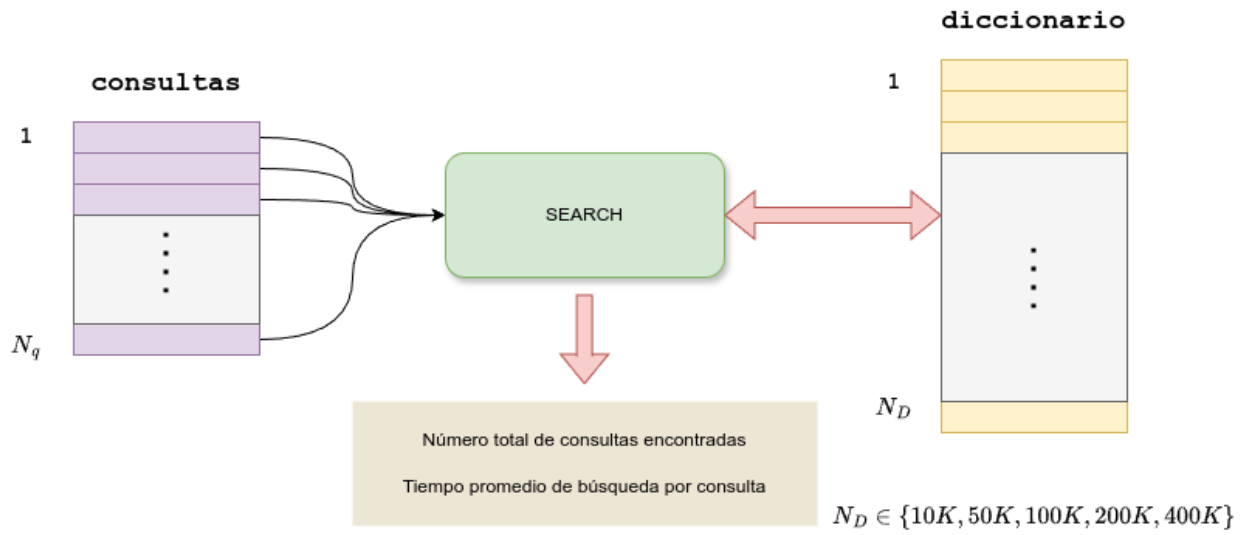


Figura 1: Esquema general de la tarea.

Tamaño ( $N_D$ )	Tiempo de Ejecución (ms)		
	AS	ABB	ABB + Sort
10K			
50K			
100K			
200K			
400K			

Cuadro 1: Ejemplo de estructura para las tablas de resultados.

- Además, deberás mostrar una gráfico (plot) mostrando el comportamiento de los dos algoritmos para diferentes valores de  $N_D$ . Un ejemplo de gráfico se muestra en Fig. 2.

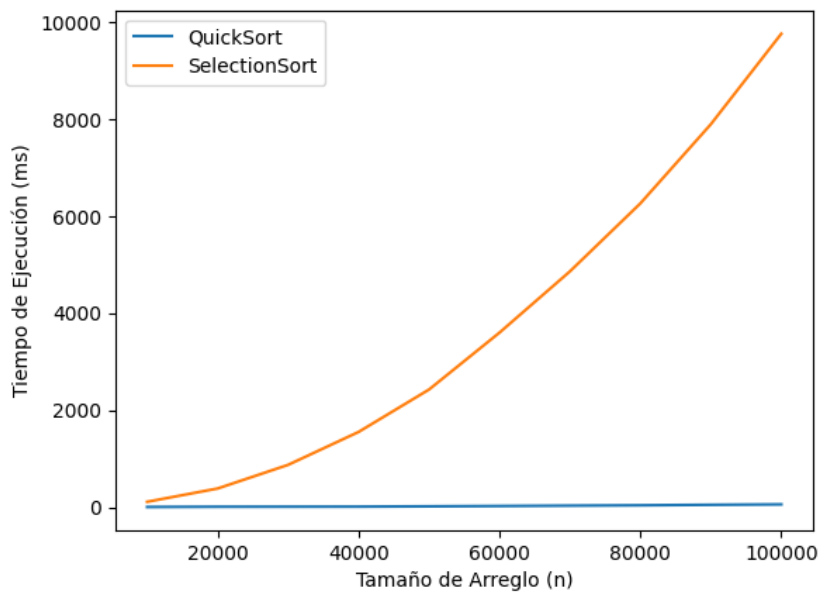


Figura 2: Ejemplo del comportamiento de dos métodos de ordenación, con respecto al tiempo de ejecución real para diversos tamaños de arreglos.

## 2.2. Datos

Los datos para la experimentación pueden ser descargados de [https://www.dropbox.com/scl/fi/hldwchbuqgdycakb1w470/t1\\_eda.zip?rlkey=o6v1vj51srsofn1lvvozhj90j&st=8fd5k4ft&dl=0](https://www.dropbox.com/scl/fi/hldwchbuqgdycakb1w470/t1_eda.zip?rlkey=o6v1vj51srsofn1lvvozhj90j&st=8fd5k4ft&dl=0).

Luego de descomprimir el archivo zip, encontrarás diversos archivos nombrados como **D<N>.txt**. Estos archivos corresponden a los diccionarios con diferentes tamaños. Además, encontrarás el archivo **Nc.txt** con 1MM de consultas, este será utilizado para realizar consultas en el diccionario.

## 3. Informe

1. **Abstract o Resumen:** es el resumen del trabajo.
2. **Introducción:** aquí se describe cada uno de los algoritmos implementados y evaluados (10 %)
3. **Desarrollo:** aquí se describe el diseño e implementación de los programas necesarios para realizar sus experimentos. Además, presenta una descripción y ejemplos de cómo usar tus programas. En esta sección es importante justificar la selección del algoritmo de ordenación. (40 %)
4. **Resultados Experimentales y Discusión:** aquí se presentan los resultados, pero lo más importante es analizarlos. Se debe observar y describir el comportamiento de los métodos. Es altamente importante relacionar los tiempos reales con los teóricos.  
**Por favor, en esta sección deben incluir las características de la computadora** sobre la que realizaron los experimentos. (40 %).
5. **Conclusiones:** ideas o hallazgos principales sobre el trabajo. (10 %)

## 4. Restricciones

1. Pueden trabajar en grupos de 2 estudiantes.
2. Todos los programas deben ser propios, permitiendo solamente utilizar el código disponible en el repositorio del curso [https://github.com/jmsaavedrar/eda\\_cpp/](https://github.com/jmsaavedrar/eda_cpp/).
3. El hallazgo de plagio será penalizado con nota 1.0, para todos los grupos involucrados.
4. Todas las implementaciones deben ser realizadas en C++.
5. **La entrega del informe es obligatorio.** Un trabajo sin informe no será calificado, asignando la nota mínima igual a 1.0.

## 5. Entrega

La entrega se debe realizar por canvas hasta el domingo 25 de agosto, 2024, 23:50 hrs. La entrega debe incluir:

1. Código fuente (en C++), junto a un README con los pasos de compilación.
2. Informe