

ENUNCIADO TAREA 1

ALGORITMOS Y COMPLEJIDAD

«Más allá de la notación asintótica: Análisis experimental de algoritmos de ordenamiento y multiplicación de matrices.»

Fecha límite de entrega: 25 de abril de 2025

Equipo ALGORITMOS Y COMPLEJIDAD 2025-1

2 de abril de 2025

23:20

Versión 0.0

Índice

1. Objetivos	2
2. Especificaciones	3
2.1. Introducción (máximo 1 página)	3
2.2. Diseño y Análisis de Algoritmos (máximo 1 página)	3
2.3. Implementar el Algoritmo (máximo 1 página)	3
2.4. Experimentos (máximo 6 páginas)	4
2.5. Conclusiones (máximo 1 página)	5
3. Condiciones de entrega	6

1. Objetivos

El objetivo de esta tarea 1 es introducirnos en el Análisis experimental de Algoritmos. Para ello, se propone realizar un estudio experimental de 3 algoritmos de ordenamiento y 3 de multiplicación de matrices.

Para cada uno de los dos problemas, el de ordenar un arreglo unidimensional de números enteros, y el de multiplicar dos matrices cuadradas, existen múltiples algoritmos que los resuelven, y la mayoría, si no todos, tienen implementaciones de fácil acceso (ya se encuentran en bibliotecas de lenguajes de programación, o son fáciles de encontrar en internet). Sus complejidades teóricas son conocidas, ya que éstas son una característica fundamental a la hora de diseñar algoritmos y, en la mayoría de los casos, la motivación para diseñarlos.

El desafío en esta tarea no está en diseñar los algoritmos, ya que como se mencionó anteriormente, estos ya están diseñados y muchas veces implementados, ni tampoco en calcular o demostrar su complejidad teórica: el desafío está en estudiar su comportamiento en la práctica y relacionar este comportamiento con la complejidad teórica.

2. Especificaciones

En esta sección se presentará la estructura que tiene que tener su informe, en particular, se detallarán las secciones que debe contener y los elementos que deben ser incluidos en cada una de ellas. En el siguiente repositorio podrá encontrar el [Template](https://github.com/pabloalvarez/INF221-2025-1-TAREA-1) que deben utilizar en esta entrega:

<https://github.com/pabloalvarez/INF221-2025-1-TAREA-1>

- No se debe modificar la estructura del informe.
- Las indicaciones se encuentran en el archivo `README.md` del repositorio y en la plantilla de \LaTeX .

2.1. Introducción (máximo 1 página)

La introducción de este tipo de informes o reportes, tiene como objetivo principal **contextualizar los problemas que se van a analizar**, proporcionando al lector la información necesaria para entender la relevancia de los mismos.

Es fundamental que en esta sección se presenten los antecedentes de los problemas, destacando investigaciones previas o principios teóricos que sirvan como base para los análisis posteriores. Además, deben explicarse los objetivos del informe, que pueden incluir la evaluación de un algoritmo, la comparación de métodos o la validación de resultados experimentales.

2.2. Diseño y Análisis de Algoritmos (máximo 1 página)

2.2.1. Ordenamiento de un arreglo unidimensional de números enteros

2.2.2. Multiplicación de matrices cuadradas de números enteros

2.3. Implementar el Algoritmo (máximo 1 página)

Cada uno de los algoritmos, para cada problema, **debe ser implementado en C++**, siguiendo los siguientes puntos:

2.3.1. Ordenamiento de un arreglo unidimensional de números enteros

Entrada:

- Leer un arreglo unidimensional A desde el archivo `{n}_{t}_{d}_{m}_1.txt` y `{n}_{t}_{d}_{m}_2.txt`, respectivamente.
 - n hace referencia a la dimensión de la matriz (n filas y n columnas) y pertenece al conjunto $\mathcal{N} = \{10^1, 10^3, 10^5, 10^7, 10^9\}$.
 - t hace referencia al tipo de matriz, y pertenece al conjunto $\mathcal{T} = \{\text{ascendente}, \text{descendente}, \text{aleatorio}\}$.
 - d hace referencia al dominio de cada elemento del arreglo $d = \{D1, D10\}$, donde $D10 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ y $D100 = \{0, 1, 2, 3, \dots, 99\}$.
 - m hace referencia a la muestra aleatoria (o caso de prueba) y pertenece al conjunto $\mathcal{M} = \{a, b, c\}$.

Salida:

- Escribir la matriz resultante $M_1 \times M_2$ en el archivo `{n}_{t}_{d}_{m}_out.txt`.

2.3.2. Multiplicación de matrices cuadradas de números enteros

Entrada:

- Leer dos matrices cuadradas de entrada M_1 y M_2 desde los archivos `{n}_{t}_{d}_{m}_1.txt` y `{n}_{t}_{d}_{m}_2.txt`, respectivamente.
 - n hace referencia a la dimensión de la matriz (n filas y n columnas) y pertenece al conjunto $\mathcal{N} = \{2^4, 2^6, 2^8, 2^{10}\}$.

- t hace referencia al tipo de matriz, y pertenece al conjunto $\mathcal{T} = \{\text{dispersa, diagonal, densa}\}$.
- d hace referencia al dominio de cada coeficiente de la matriz $d = \{D0, D10\}$, donde $D0 = \{0, 1\}$ y $D10 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
- m hace referencia a la muestra aleatoria (o caso de prueba) y pertenece al conjunto $\mathcal{M} = \{a, b, c\}$.

Salida:

- Escribir la matriz resultante $M_1 \times M_2$ en el archivo `{n}_{t}_{d}_{m}_out.txt`.

Aquí deben explicar la estructura de sus programas haciendo referencias a los archivos y funciones de su entrega. No adjunte código en esta sección.

2.4. Experimentos (máximo 6 páginas)

En la sección de Experimentos, es fundamental detallar la infraestructura utilizada para asegurar la reproducibilidad de los resultados, un principio clave en cualquier experimento científico. Esto implica especificar tanto el hardware (por ejemplo, procesador Intel Core i7-9700K, 3.6 GHz, 16 GB RAM DDR4, almacenamiento SSD NVMe) como el entorno software (sistema operativo Ubuntu 20.04 LTS, compilador g++ 9.3.0, y cualquier librería relevante). Además, se debe incluir una descripción clara de las condiciones de entrada, los parámetros utilizados y los resultados obtenidos, tales como tiempos de ejecución y consumo de memoria, que permitan a otros replicar los experimentos en entornos similares. *La replicabilidad es un aspecto crítico para validar los resultados en la investigación científica computacional [1].*

2.4.1. Datasets (máximo 2 páginas)

Diseña al menos **cinco casos de prueba** que cubran varios escenarios, incluyendo:

- Casos donde las cadenas están vacías.
- Casos con caracteres repetidos.
- Casos donde las transposiciones son necesarias.

Documentación de los Casos de Prueba:

- Proporciona las cadenas de entrada y los costos para cada operación.
- Muestra la salida esperada para cada caso de prueba, incluyendo la secuencia de operaciones y el costo total.
- Explica por qué la salida es correcta.

Es importante generar varias muestras con características similares para una misma entrada, por ejemplo, variando tamaño del input dentro de lo que les permita la infraestructura utilizada en este informe, con el fin de capturar una mayor diversidad de casos y obtener un análisis más completo del rendimiento de los algoritmos.

Aunque la implementación de los algoritmos debe ser realizada en C++, se recomienda aprovechar otros lenguajes como Python para automatizar la generación de casos de prueba, ya que es más amigable para crear gráficos y realizar análisis de los resultados. Python, con sus bibliotecas como `matplotlib` o `pandas`, facilita la visualización de los datos obtenidos de las ejecuciones de los distintos algoritmos bajo diferentes escenarios.

Además, debido a la naturaleza de las pruebas en un entorno computacional, los tiempos de ejecución pueden variar significativamente dependiendo de factores externos, como la carga del sistema en el momento de la ejecución. Por lo tanto, para obtener una medida más representativa, siempre es recomendable ejecutar múltiples pruebas con las mismas características de entrada y calcular el promedio de los resultados.

2.4.2. Resultados de los Experimentos (máximo 4 páginas)

En esta sección, los resultados obtenidos, como las gráficas o tablas, deben estar respaldados por los datos generados durante la ejecución de sus programas. Es fundamental que, junto con el informe, se adjunten los archivos que contienen dichos datos para permitir su verificación. Además, se debe permitir y especificar cómo obtener esos archivos desde una ejecución en otro computador (otra infraestructura para hacer los experimentos).

No es necesario automatizar la generación de las gráficas, pero sí es imprescindible que se pueda confirmar que las visualizaciones presentadas son producto de los datos generados por sus algoritmos, aunque la trazabilidad de los datos hasta las visualizaciones es esencial para garantizar que su validez: describa cómo se generaron los datos, cómo se procesaron y cómo se visualizaron de manera que pueda ser replicado por quien lea su informe.

2.5. Conclusiones (máximo 1 página)

La conclusión de su informe debe enfocarse en el resultado más importante de su trabajo. No se trata de repetir los puntos ya mencionados en el cuerpo del informe, sino de interpretar sus hallazgos desde un nivel más abstracto. En lugar de describir nuevamente lo que hizo, muestre cómo sus resultados responden a la necesidad planteada en la introducción.

3. Condiciones de entrega

- La tarea se realizará **individualmente** (esto es grupos de una persona), sin excepciones.
- La entrega debe realizarse vía <http://aula.usm.cl> en un **tarball** en el área designada al efecto, en el formato **tarea-1-rol.tar.gz** (rol con dígito verificador y sin guión).
Dicho **tarball** debe contener las fuentes en \LaTeX (al menos **tarea-1.tex**) de la parte escrita de su entrega, además de un archivo **tarea-1.pdf**, correspondiente a la compilación de esas fuentes.
- Si se utiliza algún código, idea, o contenido extraído de otra fuente, este **debe** ser citado en el lugar exacto donde se utilice, en lugar de mencionarlo al final del informe.
- Asegúrese que todas sus entregas tengan sus datos completos: número de la tarea, ramo, semestre, nombre y rol. Puede incluirlas como comentarios en sus fuentes \LaTeX (en \TeX comentarios son desde % hasta el final de la línea) o en posibles programas. Anótese como autor de los textos.
- Si usa material adicional al discutido en clases, detállelo. Agregue información suficiente para ubicar ese material (en caso de no tratarse de discusiones con compañeros de curso u otras personas).
- No modifique `preamble.tex`, `tarea_main.tex`, `condiciones.tex`, estructura de directorios, nombres de archivos, configuración del documento, etc. Sólo agregue texto, imágenes, tablas, código, etc. En el códigos fuente de su informe, no agregue paquetes, ni archivos `.tex` (a excepción de que agregue archivos en `/tikz`, donde puede agregar archivos `.tex` con las fuentes de gráficos en TikZ).
- La fecha límite de entrega es el día **10 de noviembre de 2024**.

NO SE ACEPTARÁN TAREAS FUERA DE PLAZO.

- Nos reservamos el derecho de llamar a interrogación sobre algunas de las tareas entregadas. En tal caso, la nota de la tarea será la obtenida en la interrogación.

NO PRESENTARSE A UN LLAMADO A INTERROGACIÓN SIN JUSTIFICACIÓN PREVIA SIGNIFICA AUTOMÁTICAMENTE NOTA 0.

Referencias

- [1] Jorge Fonseca y Kazem Taghva. «The State of Reproducible Research in Computer Science». En: ene. de 2020, págs. 519-524. ISBN: 978-3-030-43019-1. DOI: [10.1007/978-3-030-43020-7_68](https://doi.org/10.1007/978-3-030-43020-7_68).