

# Tarea 1 Algoritmos y Complejidad

PDF por Adán Contreras

Fecha entrega: Sábado 7 de Septiembre, 00:00

## 1 Algoritmos Dividir y Conquistar

Este primer informe persigue que los estudiantes realicen una evaluación experimental de distintos algoritmos de ordenamiento y de multiplicación de matrices. Así, se busca que los estudiantes sean capaces de implementar algunos de los algoritmos más conocidos para este tipo de problemas, además de contrastar su funcionamiento con implementaciones provistas por los lenguajes de programación. Adicionalmente, esta primera tarea permitirá a los estudiantes familiarizarse con herramientas de profiling y con la creación de códigos que les permitan medir las características solicitadas.

La entrega de esta tarea consistirá de un informe escrito (en formato .pdf) y los códigos asociados a los experimentos reportados en el informe. Estos códigos deben quedar públicos en algún repositorio como bitbucket o github, y estar correctamente referenciados dentro del informe.

Los algoritmos mínimos a ser considerados para los benchmarks de ordenamiento son:

- Un algoritmo cuadrático de ordenamiento (selection sort, bubble sort, insertion sort u otro)
- Mergesort
- Quicksort
- Función de sorting implementada en la biblioteca estándar del lenguaje de programación

Los algoritmos mínimos a ser considerados para los benchmarks de multiplicación de matrices son:

- Algoritmo iterativo cúbico tradicional
- Algoritmo iterativo cúbico optimizado para mantener la localidad de los datos (transponiendo la segunda matriz)
- Algoritmo de Strassen

Los juegos de datos (Datasets) deberán ser generados por cada estudiante, utilizando el método que considere oportuno. Se aconseja generar conjuntos de datos diversos que varíen en tamaño y características (e.g. semi ordenados, parcialmente ordenados, multiplicación de matrices cuadradas de la misma dimensión vs matrices de distintas dimensiones, etc). Los juegos de datos deben ser guardados en archivos (binario o texto) en un formato estándar. Cada algoritmo evaluado deberá tomar la entrada en el formato estándar especificado e imprimir el resultado en un formato de salida, también estándar y también definido por el estudiante.

## 1.1 SECCIONES SOLICITADAS:

1. Introducción: Breve reseña de todo el trabajo realizado. Descripción a alto nivel de los problemas y algoritmos (implementados y provistos por bibliotecas), herramientas, fuentes de datos y conclusiones preliminares.
2. Descripción de los algoritmos a ser comparados: Entrega de todos los códigos como adjuntos al documento o enlaces a dónde encontrarlos en el mismo. El código debe estar documentado y seguir algún estándar de codificación adoptado por el estudiante. Para cada algoritmo, especificar una descripción general y consignar el costo del mejor y peor caso. Reseñar las funciones de bibliotecas estándar utilizadas ¿Qué algoritmos funcionan por debajo y cuáles son sus costos?
3. Descripción de los datasets: En esta sección se pide listar los conjuntos de datos con los que se ha trabajado, así como caracterizarlos (tamaño del dataset, valores límites u otras características consideradas). De igual forma, se pide especificar el tipo de archivos utilizado para la entrada y salida, así como el formato de los mismos.
4. Resultados experimentales: En esta sección se deberán mostrar los resultados comparativos de los distintos algoritmos en forma de tablas, gráficos y comentar, dentro del texto, observaciones que consideren que vale la pena resaltar en las tablas/gráficos.
5. Conclusiones: En esta sección se deberán consignar observaciones/comentarios/sentencias que evalúen el trabajo realizado, así como los resultados obtenidos. En especial, se aconseja abundar alrededor del poder predictivo del análisis asintótico del peor/mejor caso y sobre la importancia de aspectos relacionados con la implementación (i.e. algoritmos inplace vs no-inplace, preservación de la localidad de los datos, otros) en los resultados experimentales.