



Complejidad de Algoritmos – Laboratorio 2

1. Objetivos

El objetivo de este laboratorio es de contextualizar y aplicar algunos de los algoritmos de búsqueda y clasificación (ordenamiento) estudiados en clase.

El laboratorio se trabajará en grupos de 2 o 3 alumnos, entregando un resultado (informe y programas) por grupo.

2. Problemas

1. En el archivo “archivo_1.tex”, encontrar la secuencia de 4 dígitos consecutivos más frecuente en el número decimal dado y decir cuantas veces se repite. Si hay más de una secuencia que se repite con esta misma frecuencia, hacer un listado de todas.
2. En el archivo “archivo_2.tex”, encontrar toda secuencia de 4 “dígitos” (hexadigitos) consecutivos que se repiten en el número hexadecimal dado.
3. En el archivo “archivo_3.tex”, cortar el número decimal en bloques de 6 dígitos consecutivos (sin repetir los dígitos) y encontrar todas las secuencia de 6 dígitos consecutivos que se repiten.

Observación: los archivos “.tex” contienen los datos que analizar, sin formateo de texto. Los programas deben ser capaz de analizar otros archivos parecidos.

3. Se solicita

1. Resolver los 3 problemas enunciados, programando las funciones necesarias para calcular la soluciones de manera eficiente
2. Entregar un informe, idealmente escrito en LaTeX, detallando las soluciones encontradas, las estrategias (algoritmos) utilizadas para resolver los problemas y justificando la elección de las estrategias.
3. Los argumentos para la justificación deben ser en contexto de complejidad asintótica (tiempo y memoria) en base a las entradas. Por ejemplo: en base a la relación entre la cantidad y de valores a analizar y la cantidad de valores posibles, el formato de los valores, etc.
4. Entregar los programas utilizados, bien escritos y documentados, idealmente en lenguaje C o C++.



4. Evaluación

La nota del laboratorio se calculará según la ponderación siguiente:

- Soluciones obtenidas [**40 %**]:
Se entregan soluciones correctas para cada pregunta del laboratorio.
- Informe [**40 %**]:
El informe describe los algoritmos elegidos para cada problema y justifica la selección de cada uno.
- Implementación [**20 %**]
El programa está escrito de forma que puede ser leído y/o re-utilizado fácilmente por otros programadores: la redacción es limpia (con espacios y divisiones claras) y bien documentada, las sub-funciones y las variables tienen nombres naturales (que indican a que sirven) o acompañadas de comentarios aclarando a que sirven.