

TAREA 2 - ELO320 ESTRUCTURA DE DATOS Y ALGORITMOS

NICOLÁS GÁLVEZ, WERNER CREIXELL
CAMPUS SAN JOAQUÍN, UTFSM. 9 DE JULIO DE 2021

1. Discordia Server

Discordia es la red de mensajería predilecta por los diferentes estudiantes de ELO320 - Estructura de Datos y Algoritmos. En ella, estudiantes se comunican en canales de voz y texto entre sus pares. Una de las grandes características es el envío de imágenes. Este servicio trabaja con imágenes de formato **.pmb**. Una imagen PBM (Portable Bit Map) es un archivo de texto que contiene una imagen monocromática cuyo formato aparece en la Figura 1. En el archivo un 1 es un píxel en negro y un 0 corresponde a un píxel en blanco.

```
P1
24 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
0 1 1 1 0 0 0 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Figura 1: Formato pbm. ¿Qué imagen contiene este archivo?

La primera línea del archivo indica el formato de imagen (P1 \rightarrow Monocromo), la segunda línea indica el tamaño de la matriz de bits en cantidad de columnas y filas (24×7), y el resto indica los valores existentes dentro de cada posición de la matriz.

Para el envío, la imagen del remitente es es encriptada/cifrada como un arreglo de números enteros, cuyos elementos están en una posición aleatoria, y almacenada en un archivo de texto **binario**. Luego el receptor, debe desencriptar el mensaje para reproducir la imagen, vale decir, leer el archivo de texto binario para obtener el arreglo de enteros, y luego ordenarlo ascendentemente para ocupar operador módulo y así obtener bits (0's y 1's) en cada posición.

Note que los archivos binarios, son archivos de texto plano que **no son** legibles por el humano, por lo tanto su formato debe ser conocido de antemano. En este caso cada línea contiene un número entero.

Además, Discordia trabaja con un formato de imagen de 1920×1080 , por lo tanto trabaja con más de **dos millones** de números enteros por imagen.

Ud. es contratado por Discordia para ayudarle a mejorar la implementación del algoritmo que descifra imágenes **.pmb** y entregar un reporte del por qué de su decisión. Se le pide:

- (20 points) Leer un archivo binario que contiene la secuencia de enteros a ordenar. El nombre del archivo debe ser entregado como parámetro del ejecutable.
- (30 points) Elegir tres algoritmos para ordenar ascendentemente el arreglo de números. Estos deben poseer las siguientes complejidades, sin repetirse:

- $\mathcal{O}(n^2)$,
 - $\mathcal{O}(n \log n)$, y
 - $\mathcal{O}(d \times n)$;
- (c) (20 points) Transformar la secuencia de valores ordenados a una imagen formato `.pbm` de tamaño 1920×1080 , utilizando el operador módulo, y verifique si genera una imagen como la mostrada en la Figura 1.
- (d) (30 points) Generar un reporte, en el archivo de texto plano `analisis.txt`, que mida empíricamente la complejidad de los algoritmos seleccionados. Esta labor la puede realizar de diversas formas:
- Midiendo el tiempo que toman cada uno de los algoritmos, a través de funciones de la biblioteca `<time.h>`.
 - Contando la cantidad instrucciones realizadas en ejecución por su algoritmo. Considere que puede escoger los siguientes escenarios: optimista, real y/o pesimista.

2. REGLAS DE ENTREGA Y CONSIDERACIONES GENERALES

- Este trabajo debe realizarse **individualmente**, vale decir, **grupos de un estudiante**. No se harán excepciones.
- Dos archivos encriptados de imágenes se encontrarán disponibles en AULA USM. Todos las imágenes fueron obtenidas de **Google Images**.
- El programa debe ser desarrollado en lenguaje C, y compilado con la versión de gcc disponible en el servidor Aragorn¹: gcc 4.8.5.
- La tarea debe ser entregada en la plata forma **AULA USM**, el día 28 de Julio de 2021 hasta las 23:59:59 Hora de Chile Continental (UTC -4).
- La tarea debe incluirse en un archivo comprimido **.tar.gz**. El nombre del archivo debe seguir la siguiente estructura: `tarea2-eda-nombre-apellido-paralelo.tar.gz`; e.g., `tarea2-eda-oliver-atom-p200.tar.gz`.
- El archivo comprimido debe incluir lo siguiente:
 - Cabecera `.h`: Archivo cabecera en el cual se deben incluir todas las bibliotecas a usar en el programa, además de las definiciones de macros, variables globales, tipos de datos personalizados, struct y prototipos de todas las funciones.
 - Código `.c`: Código del programa solicitado. Considere que éste es un desafío de al menos tamaño mediano, por lo tanto es recomendable dividir el código en diferentes archivos agrupados por su finalidad.
 - **README**: Archivo de texto plano en el cual se debe incluir: (1) una pequeña reseña del programa, (2) las condiciones de compilación y ejecución, (3) las instrucciones de compilación y ejecución, y (4) la información del creador del sistema.
 - **Makefile**: Archivo de compilación automática del sistema. Una cápsula de video para su confección se encontrará disponible en AULA USM.
- La revisión de los sistemas se hará utilizando diferentes archivos de los entregados en AULA USM. Por lo tanto, debe programar de forma genérica para cualquier cantidad de información de entrada.
- Cada fuga de memoria será penalizada. Utilice **valgrind** para verificar la correcta asignación, uso y liberación de memoria en la ejecución de su programa.
- Si su programa no compila, su nota será automáticamente **un cero (0)**.
- Cualquier atisbo de copia, será penalizada con máxima severidad.

¹`ssh aragorn.elo.utfsm.cl -l cuentausm`