**Análisis Detallado del Problema:**

El desafío propuesto por Informa2 se centra en la reconstrucción de una imagen BMP de 24 bits que ha sido sometida a múltiples transformaciones a nivel de bits y enmascaramientos. Para abordar este problema, es fundamental comprender a fondo cada uno de los componentes y procesos involucrados:

* **Imagen BMP de 24 bits:**
  + Cada píxel se compone de 3 bytes (24 bits), representando los canales RGB (rojo, verde, azul).
  + La manipulación a nivel de bits implica alterar directamente estos valores binarios, lo que puede resultar en cambios significativos en la imagen.
* **Transformaciones a Nivel de Bits:**
  + Las operaciones de XOR con la imagen de distorsión (IM) alteran los valores de los píxeles de manera reversible, siempre que se conozca la imagen IM.
  + Los desplazamientos y rotaciones de bits modifican la posición de los bits dentro de cada byte, cambiando los valores de los píxeles de forma más compleja.
  + El orden desconocido de estas transformaciones añade una capa de complejidad al problema.
* **Enmascaramiento:**
  + La suma de una porción de la imagen transformada con una máscara (M) introduce una alteración adicional que debe ser revertida.
  + El desplazamiento aleatorio (semilla) de la porción utilizada en el enmascaramiento requiere el análisis de los archivos de rastreo para determinar la posición exacta.
  + La clave de este proceso es que el enmascaramiento genera los archivos de texto, por lo tanto, la reversibilidad depende de estos.
* **Archivos de Rastreo:**
  + Estos archivos proporcionan información crucial sobre el enmascaramiento aplicado en cada etapa.
  + El desplazamiento (semilla) y los valores RGB resultantes de la suma son esenciales para revertir el enmascaramiento.

**Pasos a seguir para la Resolución:**

1. **Análisis de los Archivos de Rastreo:**
   * Extraer el desplazamiento y los valores RGB de cada archivo.
   * Determinar el tamaño de la máscara (M) y la porción de la imagen transformada utilizada en cada enmascaramiento.
2. **Reversión del Enmascaramiento:**
   * Utilizar el desplazamiento y los valores RGB extraídos para restar la máscara de la imagen transformada.
   * Reconstruir la imagen transformada antes del enmascaramiento.
3. **Deducción del Orden de las Transformaciones:**
   * Analizar los resultados de la reversión del enmascaramiento y la imagen final para identificar patrones y relaciones.
   * Experimentar con diferentes combinaciones de operaciones a nivel de bits (XOR, desplazamientos, rotaciones) para encontrar el orden correcto.
4. **Reversión de las Transformaciones a Nivel de Bits:**
   * Aplicar las operaciones inversas (XOR con IM, rotaciones y desplazamientos inversos) en el orden inverso al que se aplicaron originalmente.
   * Para la operación XOR la inversa es la misma operación. Para las rotaciones y desplazamientos, la cantidad de bits a mover es la misma, pero la dirección es opuesta.
5. **Reconstrucción de la Imagen Original:**
   * Verificar que la imagen resultante coincida con la imagen original (IO).

**Solución propuesta:**La solución propuesta se basa en un enfoque de ingeniería inversa, donde se analizan los datos proporcionados para deducir el algoritmo de transformación y revertir sus efectos. A continuación, se describen los componentes clave de esta propuesta:

* **Módulo de Análisis de Archivos de Rastreo:**
  + Este módulo se encarga de leer y procesar los archivos de rastreo, extrayendo el desplazamiento y los valores RGB.
  + Utiliza arreglos dinámicos para almacenar esta información de manera eficiente.
* **Módulo de Reversión de Enmascaramiento:**
  + Este módulo implementa el algoritmo para restar la máscara de la imagen transformada, utilizando el desplazamiento y los valores RGB extraídos.
  + Utiliza punteros para acceder y modificar los valores de los píxeles en la imagen.
* **Módulo de Deducción del Orden de Transformaciones:**
  + Este módulo implementa un algoritmo de búsqueda exhaustiva o heurística para encontrar el orden correcto de las operaciones a nivel de bits.
  + Experimenta con diferentes combinaciones y evalúa los resultados utilizando métricas de comparación de imágenes.
* **Módulo de Reversión de Transformaciones a Nivel de Bits:**
  + Este módulo implementa las funciones para aplicar las operaciones inversas (XOR, rotaciones, desplazamientos) en el orden correcto.
  + Estas funciones haran uso de operaciones a nivel de bits directamente.
* **Módulo de Reconstrucción de Imagen:**
  + Este módulo reconstruye la imagen original a partir de los datos revertidos y la exporta como un archivo BMP.
* **Interfaz Gráfica de Usuario (GUI):**
  + Se utiliza el framework Qt para crear una GUI que permita al usuario cargar los archivos de entrada (imagen transformada, imagen de distorsión, máscara, archivos de rastreo) y visualizar la imagen reconstruida.

**Consideraciones Adicionales:**

* La eficiencia del algoritmo de deducción del orden de transformaciones es crucial, ya que el número de combinaciones posibles puede ser elevado.
* La validación de la imagen reconstruida se puede realizar comparándola con la imagen original proporcionada en los conjuntos de datos de prueba.
* La implementación debe cumplir con los requisitos mínimos especificados, incluyendo el uso de punteros, arreglos dinámicos y la no utilización de STL.
* La claridad del video explicativo, al igual que los commits regulares, son muy importantes para la demostracion del entendimiento del problema.

Link del repositorio: https://github.com/nikolas3312/desafio1\_2025