



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2133 - ESTRUCTURAS DE DATOS Y ALGORITMOS

# Tarea 0

17 de agosto de 2016

2º semestre 2016 - Profesor Yadrán Eterovic

Andrés Ríos Stange - 14631636

---

## Análisis de Complejidad del Algoritmo

### Análisis del Tiempo de Ejecución

El algoritmo que aplica el filtro jéster corresponde al siguiente trozo de código:

```
45 // Aplicamos el filtro pixel a pixel
46 for (int x = 0; x < new_image->width; x++) {
47     for (int y = 0; y < new_image->height; y++) {
48         Pixel_rgb *new_pixel = get_pixel(new_image, x, y);
49         Pixel_rgb *pixel_r = get_pixel(image, x + dx, y + dy);
50         Pixel_rgb *pixel_g = get_pixel(image, x, y);
51         Pixel_rgb *pixel_b = get_pixel(image, x - dx, y - dy);
52         new_pixel->R = pixel_r->R;
53         new_pixel->G = pixel_g->G;
54         new_pixel->B = pixel_b->B;
55     }
56 }
```

Sea  $N$  el ancho de la imagen y  $M$  su alto. El algoritmo no es más que dos *for* anidados, iterando  $N$  veces en el *for* externo, y  $M$  veces en el *for* interno. Dentro del *for* interno se siguen una serie de instrucciones que toman tiempo constante  $c_0$ . Así, todo el algoritmo toma  $T(N, M) = c_0 * N * M$  pasos.

Como  $T(N, M) \in O(N * M)$ , el orden de complejidad del algoritmo es  $O(N * M)$  con  $N$  y  $M$  siendo las dimensiones de la imagen.

### Análisis del Uso de Memoria

La cantidad de memoria utilizada en el Stack es constante, por lo que no juega un papel muy importante en el análisis. Llamemos a esta constante  $c_0$ .

La memoria utilizada en el Heap son las dos imágenes: La imagen fuente y la imagen destino. Ambas imágenes tienen las mismas dimensiones. Nuevamente, sea  $N$  el ancho de las imágenes y  $M$  su alto.

La imagen se compone de  $N$  arreglos de  $M$  pixeles cada uno. Sea  $c_1$  el espacio en memoria ocupado por un pixel (generalmente  $c_1 = 3$  bytes). Entonces, cada imagen ocupa  $c_1 * N * M$  de espacio en memoria.

Así, el uso de memoria del programa es  $MEM(N, M) = 2 * c_1 * N * M + c_0 = c_3 * N * M + c_0$ .

Como  $MEM(N, M) \in O(N * M)$ , el uso de memoria del programa es de orden  $O(N * M)$ , con  $N$  y  $M$  siendo las dimensiones de la imagen.

## **Bonus: The Future**

Dado que la aplicación del filtro jister consiste en instrucciones de tipo SIMD (Single Instruction, Multiple Data) se podrían fácilmente utilizar GPU's para paralelizar el algoritmo. De esta forma, en lugar de aplicar el filtro pixel a pixel, se aplicaría el filtro a todos los pixeles en forma simultánea.

En estricto rigor el número de pasos requerido seguiría siendo  $O(N * M)$ , pero dado que todos esos pasos se ejecutan al mismo tiempo de forma paralela se podría decir que el tiempo requerido en la práctica es  $O(1)$ .