

TAREA 1 ESTRUCTURA DE DATOS Y ALGORITMOS

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Werner Creixell

Septiembre 2021

1. AUTÓMATA CELULAR

"The cellular automata consists of a sequence of sites with values 0 or 1 on a line, with each site evolving deterministically in discrete time steps according to definite rules involving the values of its nearest neighbors"[1].

Las reglas de cada iteración pueden ser expresadas según el diagrama de la figura 1, en ella el valor 1 de una casilla está representado por negro y el 0 por blanco. En la figura 1, el grupo de tres casillas representa la casilla en cuestión mas sus vecinas. Hay $2^3 = 8$ posibles combinaciones de tres casillas y un total 256 reglas posibles. En la figura se representa la regla 30.

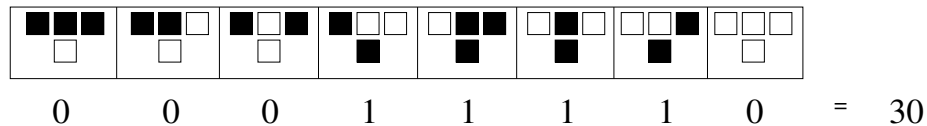


Figura 1: Regla 30

Haga un programa en C que:

1. Lea desde consola la regla a implementar (0-255) y el número de iteraciones que se ejecutará la regla y el nombre del archivo de salida (tenga presente que otro programa corregirá su tarea y usted no sabe a priori cuantas veces se ejecutará la regla). Los resultados las iteraciones deben ser almacenados en un archivo de texto de extensión .pbm ("Portable Bit Map"), que es un formato de imagen (en Unix/Linux).
2. Entregue los archivos pbm de la regla 30, ejecutada 500, 1000 y 1500 veces y el de la regla de los tres últimos dígitos de su rol (si es mayor de 255 use los dos últimos dígitos) ejecutada 500, 1000 y 1500 veces (use la condición inicial de un elemento en 1).

```

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 2: Formato pbm. ¿Qué imagen contiene este archivo?

```
werner@aragorn:~$ ./programa 30 500 werner_creixell_reg30_500.pbm
```

2. SEGMENTACIÓN DE IMÁGENES

En procesamiento digital de imágenes una tarea común es la segmentación, esto consiste en particionar la imagen en segmentos con el objetivo de delinear objetos para su posterior análisis. El método más simple de segmentación es por umbral (Thresholding en inglés), la imagen se convierte a escala de grises (cada pixel contiene un valor entre 0 y 255, donde el 0 corresponde al negro y el 255 al blanco) y posteriormente se aplica un umbral a cada pixel, si el valor del pixel es menor al umbral entonces se asigna un cero a ese pixel o un uno en caso contrario.



Figura 3: Parque nacional Nahuelbuta

La foto de la figura 3 se encuentra en aula en el formato pgm, similar al

usado en el punto anterior pero que contiene valores en escala de grises (0-255). Use este archivo para realizar la segmentación por umbral. De forma similar al programa anterior haga un programa en C que lea desde la consola el archivo fuente en pgm, el valor del umbral y el nombre del archivo de salida en formato pbm.

```
werner@aragorn:~$ ./segmentacion araucaria.pgm 150 araucaria_150.pbm
```

Pruebe su programa con los siguientes umbrales: 50, 100, 150, 200 ; y entregue un archivo pgm para cada valor del umbral. Por último, basándose en sus observaciones de éstos umbrales proponga usted el valor que considere más apropiado para esta imagen.

3. 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 30 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: **tarea1-eda-nombre-apellido.tar.gz**; e.g., **tarea1-eda-oliver-atom.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.

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

- 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.

Referencias

- [1] Stephen Wolfram. Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601–644 (1983).
- [2] Stephen Wolfram. A New Kind of Science. Wolfram Media Inc. 2002.