Instituto Politécnico Nacional

Escuela Superior de Computo

Algoritmos Genéticos

Práctica 1: Histograma

Hernández López Ángel Zait

2014080682

3CM5

Periodo: 2019/01

Planteamiento del problema
Se desea graficar una serie de datos, los cuales son tipo entero, almacenados en un arreglo unidimensional, con una capacidad de 10 elementos. Las restricciones son que se utilicen los lenguajes C o C++ para el desarrollo de la practica. El ejemplo propuesto da un arreglo de 10 elementos, el cual es el siguiente: A[10]={10,2,8,15,30,29,45,1,5,4}

Introducción
Un arreglo es un grupo o colección finita, homogénea y ordenada de elementos de un mismo tipo, en este caso, la practica solicita 10 datos de tipo entero. Un histograma es un gráfico de la representación de distribuciones de frecuencias, que por lo general, se utilizan rectangulos dentro de unas coordenadas dadas. Estas herramientas sirven en los algoritmos genéticos para graficar el comportamiento de los individuos, entre otras cosas.
Se utilizó el sistema operativo Ubuntu 14.04 LTS, ya que cuenta con el compilador de lenguaje C y C++ nativo, para crear el código, se utilizó el editor de textos Sublime Text.

Para el desarrollo de esta practica, se utilizo una biblioteca especial de C, llamada graphics.h, el cual contiene herramientas para el desarrollo de figuras geométricas, lineas, letras, entre otros. Para la instalación de la biblioteca, se descarga el contenido de la páguina: https://askubuntu.com/questions/525051/how-do-i-use-graphics-h-in-ubuntu .
Después de la descarga se introduce en la termila el siguiente comando:

```
sudo apt-get install build-essential
```

Para instalar los paquetdes adicionales, podemos poner el siguiente comando

```
sudo apt-get install libsdl-image1.2 libsdl-image1.2-dev guile-1.8 \
 guile-1.8-dev libsdl1.2debian libart-2.0-dev libaudiofile-dev \
  libesd0-dev libdirectfb-dev libdirectfb-extra libfreetype6-dev \
  libxext-dev x11proto-xext-dev libfreetype6 libaa1 libaa1-dev \
  libslang2-dev libasound2 libasound2-dev
```

Donde guardamos el archivo descargado, deberemos ir ahí y descomprimirlo, de igual forma en la terminal, dirigirlos al directorio donde está guardado el archivo descomprimido. Ya descomprimido, ingresar el siguiente comando en la terminal.

```
./configure
make
sudo make install
sudo cp /usr/local/lib/libgraph.* /usr/lib
```

Ahora se puede utilizar la biblioteca de graphics.h en el sistema operativo de Ubuntu.

Para poder utilizar los graficos incluidos en la biblioteca graphics.h debemos poner en el codigo C la inicialización del modo gráfico, el codigo es el siguiente

```
int gd = DETECT,gm;
initgraph(&gd,&gm,NULL);
```

Donde le estamos diciendo al compilador que auto detecte el driver correspondiente a nuestra tarjeta gráfica y el path (en este caso valor nulo). Y para cerrar el modo gráfico, se hace con la sentencia:

```
closegraph();
```

Para poder hacer la gratificación de los datos del arreglo se utilizaron barras, rectángulos, y

texto para tener mejor visión de los datos. A continuación, se explicara brevemente las funciones utilizadas en el código.

- rectangle(x1,y1,x2,y2): Está función crea la figura de un rectángulo, donde x1,y1,x2,y2 son números enteros, los cuales representan las coordenadas del rectángulo, y que representan el píxel en donde se encuentra dicho vértice.

```
rectangle(10,20,100,200);
```

- bar(x1,y1,x2,y2): De igual forma que rectangle, crea un rectángulo relleno, en coordenadas específicas, así que, usamos bar para hacer la barra de un dato, y rectangle para hacer un contorno cada barra.

```
bar(x1,y1,x2,y2):
```

- setcolor(a): Le asigna un color a un objeto que se encuentre dentro del modo gráfico; cuenta con una paleta de 16 colores, el dato ingresado debe ser entero y los números van del 0 al 15, un número para cada color.

```
setcolor(4);
```

- outtextxy(x,y,texto): Está función nos ayuda a crear un texto dentro del modo gráfico, donde "x y 'y' son los píxeles en donde se ubicará el texto, de tipo entero. Y 'texto' es el mensaje que se desea imprimir, el cual es de tipo string.

```
Outtextxy(10,20,”Hola mundo”);
```

- delay(time): Es un temporizador, el cual nos indica cuanto tiempo estará activo el modo gráfico, el parámetro de entrada debe ser entero y está en milisegundos.

```
delay(30000);
```

Para poner un texto a la función outtextxy() se tuvo que hacer una conversión de entero a string, ya que marcaba un error al poner un dato entero en lugar de string, así que se utilizó la función sprintf(a,b,c), donde a es la variable donde se guardará en dato string, b el tipo de dato al cual se recibirá para pasar a string, en este caso es entero, así que se puso “%d”, y c que es la variable a transformar:

```
char da[10];
sprintf(da,"%d",20);
```

Contenido

Para poder graficar el arreglo a[10]={10,2,8,15,30,29,45,1,5,4} se hizo un código en C++ donde se usan algunas funciones de graphics. Se utilizó un generador de números aleatorios, para poder ponerle un color distinto, sin que se repitiera, a las barras que representan los datos que se desean graficar.

```
int i=0,pix=20,pix2=50; char da[5];
```

Para ello el proceso de generar las barras de cada uno de los valores se utilizó un ciclo while para poder recorrer todo el arreglo, valor por valor y graficarlo. Los píxeles iniciales para el texto fueron (10,pix2)-20, para que estuviera centrado con las barras.

```
sprintf(da,"%d",i+1);
outtextxy(10,(pix2-20),da);
```

Para la barra se inicializó con los píxeles (30,20) y se finalizaron en el valor del primer dato, el cual, se le multiplico por 10 y se le sumaron 30 (y), esto para que los valores menores de 30 no se empalmaran con el texto que ya está ocupando las coordenadas (10,30), de igual forma, para los rectángulos, tienen las mismas coordenadas que las barras, esto para que funcionara como el contorno de cada barra, y se distinguiera un poco más el contenido.

```
y=(a[i]*10)+30;
bar(30,pix,y,pix2);
```

rectangle(30,pix,y,pix2);

Cada vez que se hiciera una barra, el programa recorrería el arreglo y a su vez los píxeles, pero solamente del lado de la ordenada, ya que se graficó horizontalmente, por facilidad del uso de las coordenadas. Para el texto del valor que tiene el arreglo en esa ubicación, se le pondrá como primer coordenada el valor y

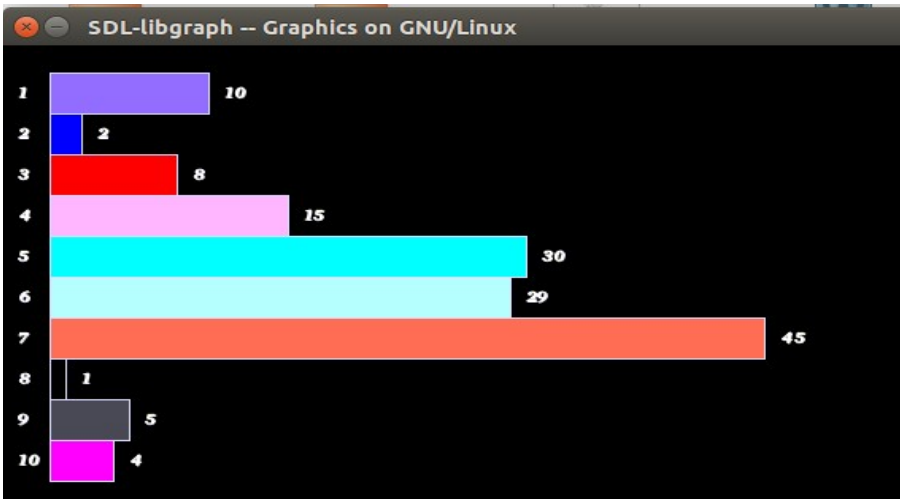
El primer valor de la ordenada (pix) sera el valor pix2 de la barra ya graficada, y el segundo valor de pix2 ahora será el nuevo valor de pix1 más 30 unidades. Lo cual nos ayuda a que las barras estén enseguida una de otra.

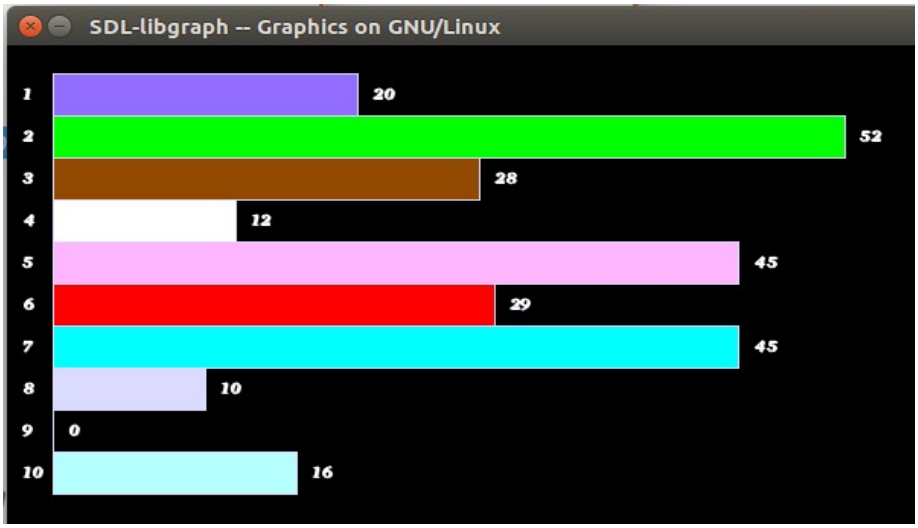
pix=pix2; pix2=pix+30;

Así hasta acabar con todos los valores del arreglo, y detener el modo gráfico aproximadamente 20 segundos, para apreciar el resultado.

Para la compilación del programa en la terminal, utilizamos el siguiente comando:

g++ nombre.cpp -o nombre -lgraph





Conclusiones

Se pudo realizar un histograma con ayuda de la biblioteca graphics de C, el cual pude ser de gran ayuda para poder desarrollar gráficos sencillos, además de saber utilizar las herramientas que nos ofrecen, amoldándolas a nuestro gusto, con fin de realizar un objetivo.