



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



GENETIC ALGORITHMS

REPORTE PRACTICA I

“HISTOGRAMA”

ALUMNO: SOLIS SANCHEZ JHOVANY

GRUPO: 3CM5

BOLETA: 2015630489

INTRODUCCIÓN

Los objetivos de esta primera practica de algoritmos genéticos serán decidir el sistema operativo que se utilizara, configurar y conocer el uso de las librerías necesarias en el lenguaje de programación C/C++ para graficar los resultados obtenidos por los algoritmos implementados, así como de configurar el entorno de desarrollo que se utilizara en futuras practicas.

La practica consistirá en realizar pruebas de funcionalidad graficando histogramas a partir de arreglos lineales.

El sistema operativo utilizado sera Linux mint el cual es una distribución de linux/debian; para la parte de la graficacion se utilizara scripts hechos en el lenguaje de programación python. los cuales contarán con las librerías matplotlib y numpy, con el fin de aprovechar la facilidad de uso que tiene este lenguaje.

DESARROLLO

La función hist recibirá como parámetros el array con los datos a graficar y el tamaño del mismo; la macro GRAPHIC_HIST se encargara a través del de la función system de ejecutar la instrucción necesaria para iniciar el script en python.

```
int main() {  
  
    int array[] = {5,8,4,7,19,8,6,15,1,5,4};  
    int tam = sizeof(array)/sizeof(int);  
  
    hist(array,tam);  
  
    GRAPHIC_HIST;  
  
    return 0;  
}
```

Illustration 1: Función hist

```
#ifndef _HISTOGRAM_A_  
#define _HISTOGRAM_A_  
  
#define GRAPHIC_HIST system("python3 graph.py")  
  
void hist(int *values, int tam);  
  
#endif // _HISTOGRAM_A_
```

Illustration 2: Macro

Resultado obtenido es graficado por matplotlib utilizando internamente la GUI de tkinter.

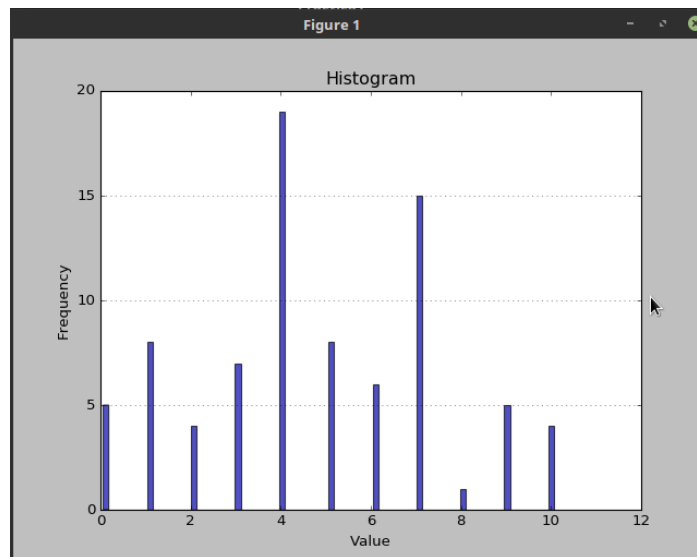


Illustration 3: Histograma obtenido

Los datos son procesados por la función hist la cual como resultado almacena un archivo txt en la carpeta temporal del sistema con los datos necesarios para que el script los pueda leer.

```
void hist(int *values, int tam){
    int i=0, j,k;
    FILE *fp;
    fp = fopen("/tmp/values.txt", "w");

    do{
        j = values[i];
        for(k=0;k<j;k++)
            fprintf(fp, "%d\n", i);

    }while(i++<tam);

    fclose(fp);
}
```

Illustration 4: Archivo temporal

Además todas las etiquetas mostradas en la grafica final pueden ser modificadas a través del código en python y en una futura mejora se podrán realizar estas configuraciones directamente desde el código fuente en C.

Finalmente el código del script en python lee el archivo desde la carpeta temporal y utiliza la función hist propia de la librería matplotlib desplegar los datos en un histograma.

```
import matplotlib.pyplot as plt
import numpy as np

def readValues():
    file = open("/tmp/values.txt")
    lines = file.readlines()
    list = []
    for x in lines:
        if x != "\n":
            list.append(int(x))

    return list

print("Creating histogram...")
values = readValues()
l = len(values)
n, bins, patches = plt.hist(x=values, bins=l, range=(0, values[l-1]), color='#0504aa',
                             alpha=0.7, rwidth=1.8, align='right')

plt.grid(axis='y', alpha=0.75)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.text(23, 45, r'$\mu=15, b=3$')
maxfreq = n.max()

plt.ylim(ymax=np.ceil(maxfreq / 10) * 10 if maxfreq % 10 else maxfreq + 10)
plt.show()
```

Illustration 5: Script de python

CONCLUSIÓN

Como se puede dar uno cuenta no se utilizo un completamente el lenguaje C para realizar las graficas, sino que se vinculo otro lenguaje para que las realizara, esto debido a que se encontraron algunas dificultades cuando se intentaba instalar algunas librerías en C en el S.O Linux. Se espera que se pueda solucionar esas dificultades o de encontrar algunas otras librerías mas eficientes completamente para el desarrollo en C/C++. Por el momento seguirá utilizando scripts de python solo para realizar la graficacion de los resultados.