|  |  |  |
| --- | --- | --- |
|  | **Instituto Politécnico Nacional**  *Escuela Superior de Cómputo* |  |
|  | **Genetic Algorithms** |  |
|  | Profesor: M. en C. Maria Elena Cruz Meza |  |

**Practica 1**

**Graficando una lista de datos generados aleatoriamente.**

Grupo: 3CM8

Objetivo: Mostrar los conocimientos en la manipulación de histogramas.

Pérez Garduño José Emiliano

*Fecha: 10 de septiembre de 2019*

Marco Teórico:

Histograma: Es una gráfica de la distribución de un conjunto de datos. Es un tipo especial de gráfica de barras, en la cual una barra va pegada a la otra, es decir no hay espacio entre las barras. Cada barra representa un subconjunto de los datos.

Un histograma muestra la acumulación ó tendencia, la variabilidad o dispersión y la forma de la distribución, es una gráfica adecuada para representar variables continuas, aunque también se puede usar para variables discretas. Es decir, mediante un histograma se puede mostrar gráficamente la distribución de una variable cuantitativa o numérica.

Desarrollo de la Práctica:

# main.c :

/\*

Codigo para generar una lista de numeros aleatorios y mostrar su histograma

Autor: José Emiliano Pérez Garduño

Materia: Genetic Algorithms

Grupo: 3CM8

\*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "practica0.h"

int main(int argc, char const \*argv[])

{

Practica0();

printf("\n\nPrograma para mostrar una lista de 10 a N numeros aleatorios junto a su histograma\n");

double \* Lista;

Lista = CrearLista(Lista);

Lista = FuncionCuadrado(Lista);

MostrarHistograma(Lista);

return 0;

}

int LongitudLista(double \* L)

{

int i = 0;

while(L[i] != 0)

i++;

return i;

}

double \* CrearLista(double \*L)

{

int n;

printf("Introducir cuantos numeros a partir de 10 tendra la lista (10 + n)\n");

scanf("%d",&n);

printf("Se creara una lista de tamano: 10 + %d = %d\n", n, 10+n);

L = calloc(sizeof(double) \* (10 + n),0);

L = PoblarLista(n,L);

return L;

}

double \* PoblarLista(int n, double \* L)

{

srand(time(NULL));

int i;

for (i = 0; i < 10+n; i++)

L[i] = rand() % 50 + 1;

MostrarLista(L);

return L;

}

void MostrarLista(double \*L)

{

printf("Lista de Numeros Aleatorios -> ");

int i;

int l = LongitudLista(L);

printf("A[%d]={",l);

for(i = 0; i < l; i++)

{

if (i == l-1)

printf("%.0lf}\n",L[i]);

else

printf("%.0lf,",L[i]);

}

}

double \* FuncionCuadrado(double \* L) //funcion para evaluar numeros enteros f(x) = x²

{

int i;

int l = LongitudLista(L);

for (i = 0; i < l; i++)

{

L[i] = L[i] \* L[i];

}

printf("\nLista de Numeros x² -> A²[%d]={",l);

for(i = 0; i < l; i++)

{

if (i == l-1)

printf("%.0lf}\n",L[i]);

else

printf("%.0lf,",L[i]);

}

fflush(stdin);

getchar();

fflush(stdin);

getchar();

return L;

}

void MostrarHistograma(double \*L)

{

printf("\nHistograma: \n\n");

int i,j,ubicacion;

int l = LongitudLista(L);

ubicacion = NumeroMaximo(l,L);

//Parte del eje Y

printf("El numero maximo es: %0.lf\n",L[ubicacion]);

printf("--------------------------------------------------\n");

for (i = L[ubicacion]; i > 0; --i) {

printf("%.3d|", i);

for (j = 0; j < l; ++j)

(L[j] >= i) ? printf(" X ") : printf(" ");

printf("\n");

}

printf(" ");

for (i = 0; i < l; ++i) //Parte del eje X

printf("%3d", i+1);

printf("\n--------------------------------------------------\n");

}

int NumeroMaximo(int l, double \*L)

{

double max;

max = L[0];

int i;

int ubicacion;

for (i = 0; i < l; i++)

{

if(L[i] > max)

{

max = L[i];

ubicacion = i;

}

}

return ubicacion;

}

int ConseguirLongitudLista(char \* L) //Función para conseguir la longitud de una lista

{

int l;

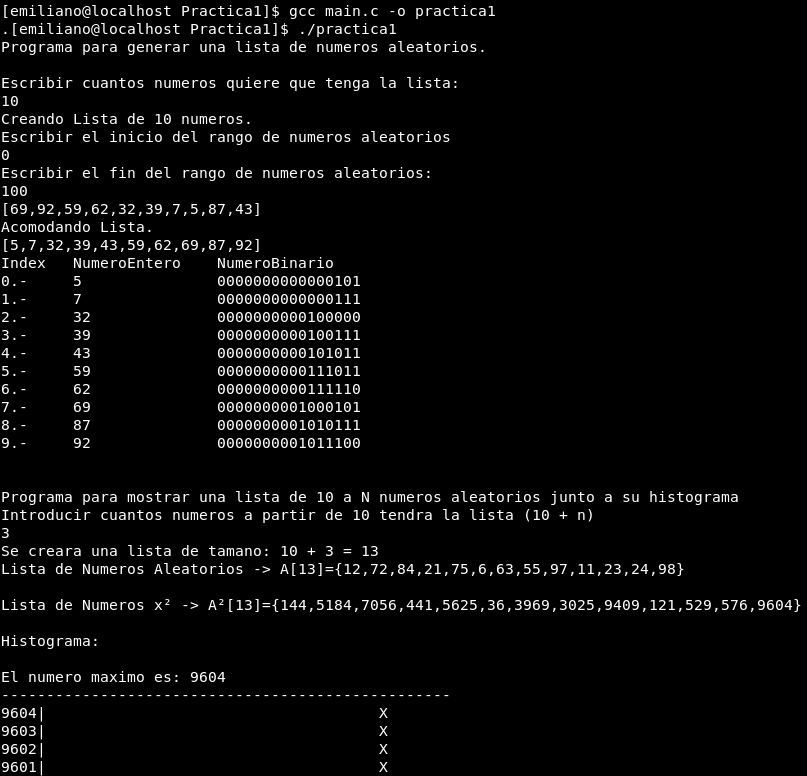
l = 0;

while (L[l] != '\0')

l++;

return l;

}

char \* PoblarListaPractica0(int l, char \* L)

{

int i, n, fin, inicio;

time\_t t;

srand((unsigned) time(&t));

printf("Escribir el inicio del rango de numeros aleatorios\n");

scanf("%d",&inicio);

printf("Escribir el fin del rango de numeros aleatorios:\n");

scanf("%d",&fin);

for (i = 0; i <= l; i++)

{

if(i == l)

{

L[i] = '\0';

}

else

{

n = rand() % (fin-inicio) + inicio;

L[i] = n;

}

}

return L;

}

void MostrarListaPractica0(char \* L)

{

int l = ConseguirLongitudLista(L);

int i;

printf("[");

for (i = 0; i < l; i++)

{

if(i != l-1)

printf("%d,",L[i]);

else

printf("%d]\n",L[i]);

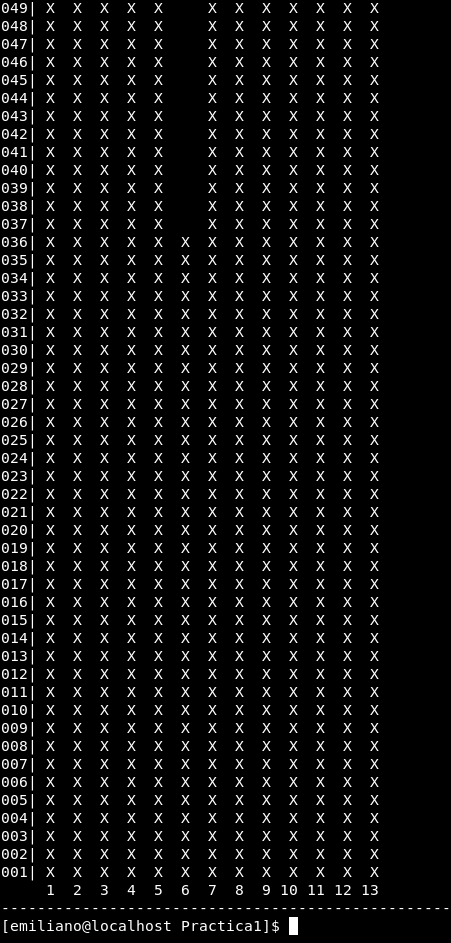
}

}

char \* AcomodarLista(char \* L)

{

int i, j, aux;

 int l = ConseguirLongitudLista(L);

printf("Acomodando Lista.\n");

for(i = 0; i < l; i++)

{

for(j = i + 1; j < l; j++)

{

if(L[i] >= L[j])

{

aux = L[i];

L[i] = L[j];

L[j] = aux;

}

}

}

MostrarListaPractica0(L);

return L;

}

void ListaBinario(char \* L)

{

int i, l;

l = ConseguirLongitudLista(L);

printf("Index\tNumeroEntero\tNumeroBinario\n");

for (i = 0; i < l; i++)

{

printf("%d.- \t%d\t\t",i,L[i]);

NumBinario(L[i]);

}

}

void NumBinario(int num)

{

int i, j;

for (i = 15; i >= 0; i--)

{

j = num >> i;

if (j & 1)

printf("1");

else

printf("0");

}

printf("\n");

}

void Practica0()

{

int lBin;

char \*LBin;

printf("Programa para generar una lista de numeros aleatorios.\n\n");

printf("Escribir cuantos numeros quiere que tenga la lista:\n");

scanf("%d",&lBin);

printf("Creando Lista de %d numeros.\n",lBin);

LBin = calloc(sizeof(int) \* (lBin+1),0);

LBin = PoblarListaPractica0(lBin,LBin);

MostrarListaPractica0(LBin);

LBin = AcomodarLista(LBin);

ListaBinario(LBin);

getchar();

}

# practica0.h :

#include <stdio.h>

int ConseguirLongitudLista(char \* L); //Función para conseguir la longitud y crear la lista

char \* PoblarListaPractica0(int l, char \* L); //Función para poblar aleatoriamente la lista

void MostrarListaPractica0(char \* L); //Función para mostrar la Lista

char \* AcomodarLista(char \* L); //Función para acomodar la Lista

void ListaBinario(char \* L); //Función para imprimir la lista en binario

void NumBinario(int n); //Función para imprimir un numero entero en binario (16 bits)

void Practica0(); //Función para realizar la practica 0

double \* CrearLista(); //Función para CrearLista de doubles

double \* PoblarLista(int n, double \* L); //Función para Poblar la lista creada

double \* FuncionCuadrado(double \* L); //Función para elevar al cuadrado a la lista

int LongitudLista(double \* L); //Función para conseguir la longitud de una lista double

void MostrarLista(double \* L); //Función para mostrar lista double

void MostrarHistograma(double \*L); //Función para imprimir histograma

int NumeroMaximo(int l, double \*L); //Función para encontrar el numero mas grande

Referencias:

[*http://asesorias.cuautitlan2.unam.mx/Laboratoriovirtualdeestadistica/DOCUMENTOS/TEMA%201/7.%20HISTOGRAMAS.pdf*](http://asesorias.cuautitlan2.unam.mx/Laboratoriovirtualdeestadistica/DOCUMENTOS/TEMA 1/7. HISTOGRAMAS.pdf)