

Programa 1 - Universo

Leon Tejeda 2CM5

Octubre 2020

Se nos dejo la tarea de hacer un programa que imprima en manera de conjunto todo el universo de cadenas binarias de un largo dado

INTRODUCCION:

Primero empezaremos definiendo que es un numero binario: Un numero binario se representa por unicamente 0's y 1's, cualquier numero se puede representar mediante este sistema de numeros.

Universo de cadenas binarias: Para entender el problema, es necesario entender que un universo de cadenas binarias, contiene a todas las cadenas binarias de todos los largos hasta el largo dado.

Esto quiere decir que se tendran $2^n + 2^{(n-1)} + 2^{(n-2)} + \dots + 2 + 1$

PLANTEAMIENTO DEL PROBLEMA:

Programar el universo de las cadenas binarias (Σ^n). Dada una "n" que introduzca el usuario o que el programa lo determine automáticamente. El rango de "n" debe de estar en el intervalo de [0,1000].

1. El programa debe de preguntar si quiere calcular otra "n" o no.
2. La salida, expresada en notación de conjunto, debe ir a un archivo de texto.
3. Una segunda salida (archivo de texto) debe concatenar todas las cadenas calculadas en una sola cadena, quitar las llaves, comas y cualquier otro símbolo que no sean 0s y 1s.
4. Del primer archivo de salida, graficar el número de 1s de cada cadena. El eje de las x es la cadena y el eje de las y el número de 1s que tiene esa cadena. Específicamente, calcular y graficar cuando $n=23$. Al mismo tiempo, calcular la gráfica pero calculando su logaritmo en base 2 y 10 respectivamente.
5. Del segundo archivo de salida, particionar la cadena en subcadenas de longitud 32 y graficar la cantidad de unos de esas subcadenas. Al mismo tiempo, calcular la gráfica pero calculando su logaritmo en base 2 y 10 respectivamente.

IMPLEMENTACION:

Implementamos un algoritmo que convierta los numeros decimales a binarios Despues rellenamos el numero con ceros dependiendo del largo necesario, para llegar al largo necesario

Se contabilizo el numero de 1's que tenia el numero y se checo el numero de la cadena, esto se guardo en un archivo como coordenadas y la cadena se guardo en un archivo como conjunto

Al graficar se tomo en cuenta el numero de 1's como las y's y el numero de la cadena como 0's

Codigo:

```
import random
import os
import math
import cmath
from matplotlib import pyplot

#Primera Grafica
def pasar_coordenadas1(x, y):
    archivo = open("Programa1_Coordenadas1.txt", "a")
    archivo.write(str(x))
    archivo.write(",")
    archivo.write(str(y))
    archivo.write("\n")
    archivo.close()

#Segunda Grafica (32 bits)
def pasar_coordenadas2(x,y):
    archivo = open("Programa1_Coordenadas2.txt", "a")
    archivo.write(str(x))
    archivo.write(",")
    archivo.write(str(y))
    archivo.write("\n")
    archivo.close()

def grafica1(k):

    archivo = open("Programa1_Coordenadas1.txt", "r")
    for coordenada in archivo:
        x = False
        y = False
        separacion = False

        for dato in coordenada:
            if dato != "," and dato != " " and separacion == False:
                if x == False:
                    x = dato

            else:
                x = x + dato

        elif dato != "," and dato != " " and separacion == True:
```

```

        if y == False:
            y = dato

        else:
            y = y + dato
    else:
        separacion = True

    pyplot.plot(int(x), int(y), marker=".", color="blue")

    # Logaritmo base 2
    if int(y) != 0:
        y2 = math.log2(int(y))
    else:
        y2=0

    pyplot.plot(int(x), y2, marker=".", color="red")

    # Logaritmo base 10
    if int(y) != 0:
        y10 = math.log10(int(y))
    else:
        y10=0

    pyplot.plot(int(x), y10, marker=".", color="green")

# Establecer el color de los ejes.
pyplot.axhline(0, color="black")
pyplot.axvline(0, color="black")

# Limitar los valores de los ejes.
pyplot.xlim(0, 2**k)
pyplot.ylim(0, k)

#Salva la Grafica
pyplot.savefig("Programa1_Grafica1.png")

# Mostrarlo.
#pyplot.show()

archivo.close()

```

```

def grafica2(k):
    #print("Aqui va la grafica2")
    archivo = open("Programa1_Coordenadas2.txt", "r")
    for coordenada in archivo:
        x = False
        y = False
        separacion = False

        for dato in coordenada:
            if dato != "," and dato != " " and separacion == False:
                if x == False:
                    x = dato

                else:
                    x = x + dato

            elif dato != "," and dato != " " and separacion == True:
                if y == False:
                    y = dato

                else:
                    y = y + dato

            else:
                separacion = True

        pyplot.plot(int(x), int(y), marker=".", color="blue")

    # Logaritmo base 2
    if int(y) != 0:
        y2 = math.log2(int(y))
    else:
        y2=0

    pyplot.plot(int(x), y2, marker=".", color="red")

    # Logaritmo base 10
    if int(y) != 0:
        y10 = math.log10(int(y))
    else:
        y10=0

    pyplot.plot(int(x), y10, marker=".", color="green")

    # Establecer el color de los ejes.
    pyplot.axhline(0, color="black")

```

```

pyplot.axvline(0, color="black")

# Limitar los valores de los ejes.
pyplot.xlim(0, 2**k)
pyplot.ylim(0, 32)

#Salva la Grafica
pyplot.savefig("Programa1_Grafica2.png")

# Mostrarlo.
#pyplot.show()
archivo.close()

#cuenta cuantos 1s hay en el numero binario
def unos(numero):

    unos = 0
    if numero >= 1:
        while numero > 1:

            residuo = numero % 2
            numero = numero // 2

            if residuo == 1:
                unos = unos + 1

        unos = unos + 1

    return unos

#Transformador (Binario-Decimal)
def trans_decimal(cadena):
    numero = 0
    posicion = len(cadena)-1

    while posicion > -1:
        aux = (len(cadena)-1) - posicion
        if cadena[posicion] == "1" and numero == 0:
            numero = 2 ** aux
        elif cadena[posicion] == "1":
            numero = numero + (2**aux)

        posicion = posicion - 1
    return numero

```

```

#Transformador (Deccimal-Binario)
def trans_binario(numero):
    binario = 0

    #print ("Transformando a binario ....")

    if numero > 1:
        while numero > 1:

            residuo = numero % 2
            numero = numero // 2

            if binario == 0:
                binario = str(residuo)

            else:
                binario = str(residuo) + binario

        binario = str(numero) + binario

    else:
        binario = str(numero)

    return binario

def rellenar(cadena, largo):

    if(largo == len(cadena)):
        return cadena
    else:
        for x in range (0, largo - len(cadena)):
            cadena = "0" + cadena

    return cadena

#Archivo 1 (Forma de Conjunto)
def pasarlo_archivo1(binario, k):
    archivo = open("Programa1_Archivo1.txt", "a")

    if k == 0:
        archivo.write("{")
        archivo.write("e")

    else:

```

```

        archivo.write(",")
        archivo.write(binario)

    archivo.close()

#Archivo2 (Forma de cadena)
def pasarlo_archivo2(binario, k):
    archivo = open("Programa1_Archivo2.txt", "a")

    if k != 0:
        archivo.write(binario)

    archivo.close()

#Archivo2 (Separacion de 32 bits)
def separador_de_32_bits_archivo2():
    contador = 0
    binario = 0
    archivo = open("Programa1_Archivo2.txt", "r")

    for linea in archivo:
        for caracter in linea:
            if contador < 31 and binario == 0:
                binario = caracter
                contador = contador + 1

            elif contador < 31:
                binario = binario + caracter
                contador = contador + 1

            else:
                contador = 0
                binario = binario + ","
                binario = binario + caracter

    archivo.close()
    binario = binario + ","
    return binario

def borrar_archivo_existente():
    archivo = open("Programa1_Archivo1.txt", "w")
    archivo.close()

    archivo = open("Programa1_Coordenadas1.txt", "w")
    archivo.close()

```

```

archivo = open("Programa1_Archivo2.txt", "w")
archivo.close()

archivo = open("Programa1_Coordenadas2.txt", "w")
archivo.close()

def universo(k):

    coordenada_x = 0
    coordenada_x1 = 1
    coordenada_y1 = 0

    for i in range(0, k+1):
        largo = 2 ** i

        for j in range(0, largo):
            #Transforma iocn a binario y Relleno de 0
            binario = trans_binario(j)
            binario = relleno(binario, i)

            #Escritura Archivo1
            pasarlo_archivo1(binario, i)

            #Coordenadas de la Primera Grafica
            coordenada_y = unos(j)
            pasar_coordenadas1(coordenada_x, coordenada_y)
            coordenada_x = coordenada_x + 1

            #Escritura Archivo2
            pasarlo_archivo2(binario, i)

    archivo = open("Programa1_Archivo1.txt", "a")
    archivo.write("}")
    archivo.close()

    aux=separador_de_32_bits_archivo2()
    archivo = open("Programa1_Archivo2.txt", "w")
    archivo.write(aux)
    archivo.close()

    #Coordenadas de la Segunda Grafica
    archivo = open("Programa1_Coordenadas2.txt", "w")
    for caracter in aux:
        if caracter == ",":

```



```

        coordenada_y1 = trans_decimal(coordenada_y1)
        coordenada_y1 = unos(int(coordenada_y1))

        pasar_coordenadas2(coordenada_x1, coordenada_y1)
        coordenada_x1 = coordenada_x1 + 1
        coordenada_y1 = 0

    elif coordenada_y1 == 0:
        coordenada_y1 = caracter

    else:
        coordenada_y1 = coordenada_y1 + caracter

archivo.close()

grafica1(k)
grafica2(k)

def menu():
    print("Seleccione una opcion")
    print("1. Ingresar un numero entre 0-1000")
    print("2. Generar un numero aleatorio entre 0-1000")

    op = int(input(""))

    if op == 1:
        print("Ingrese un numero entre 0-1000")
        k = int(input(""))
        universo(k)

    elif op == 2:
        k = random.randint(0, 1000)
        print("El numero aleatorio es igual a: " + str(k))
        universo(k)

    else:
        print("Ingrese un numero valido")

repetir = 1

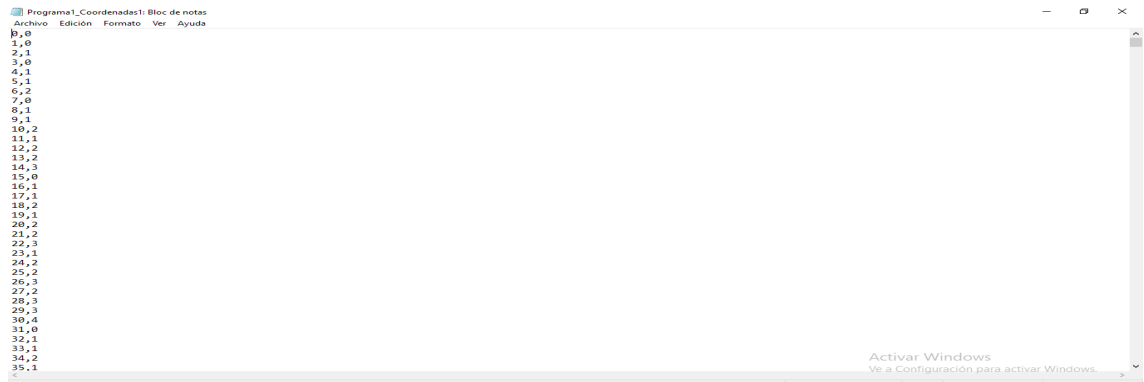
while repetir == 1:
    borrar_archivo_existente()
    menu()
    print("Si desea ingresar otro numero presione (y)")

```

Esta es la forma e la que imprime los datos en forma de conjunto

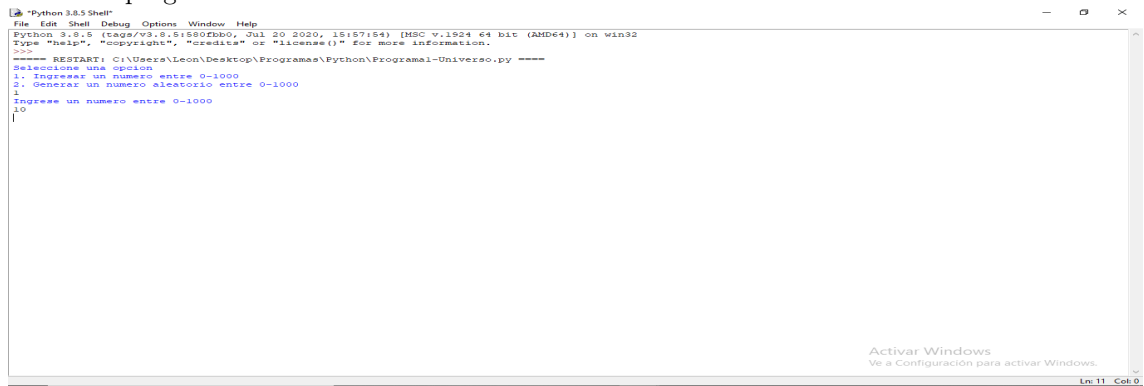
Esta es la forma e la que imprime los datos separados en 32 bits

Coordenadas por cantidad de 1's

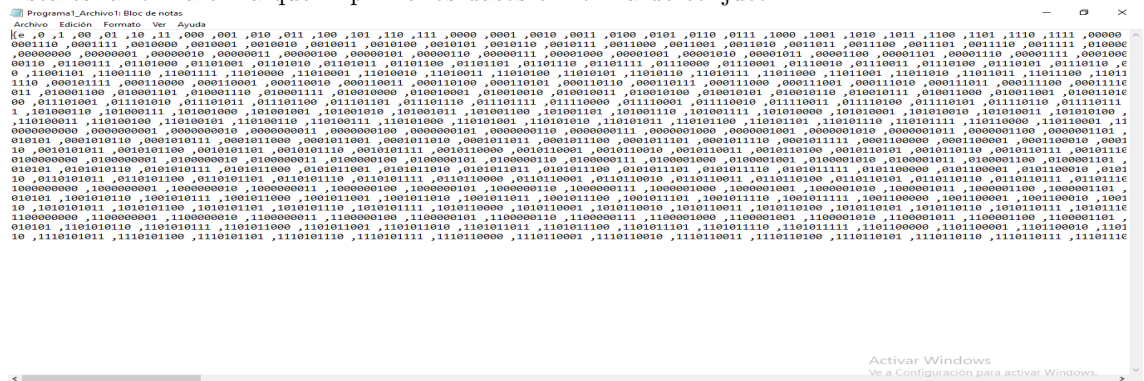


Adjuntaremos imagenes de cuando n=10 (incluyendo grafica), ya que con el valor solicitado, no puede graficarse al ser muchas coordenadas
Cuando n = 10

Menu del programa



Esta es la forma en la que imprime los datos en forma de conjunto



Esta es la forma en la que imprime los datos separados en 32 bits

```
Programa1_Archivo2: Bloc de notas
Archivo Edición Formato Ver Ayuda
01100011011000001010011100101110,1110000001001000110100010101100,111100010011010101110011011101,11100000000001000100001001000010,1001100011101000010010101010110,1100
1100100110011011010000101010101,0110011011101110000111001011010,011101011110001111010111100111,11110000010000010000101000011,000100100010100010100011110010,00100
0011100000110010011101000111,011001111000111101001111000111,111010000001000001010001001000,01101000100010001010100011001000,11101001000010010010100101001001,011010
01101001011010101010101010,111101100001011001001110101011,011101111001011110101011,11111000000110000011100001011000,0111100010011000101110001011000,1111100
000010101100010110000010110,100010111000010111000110000001,100010001100100001101100,000110101000101100001101100011,1000000111001000111011010,00111100
0011001001001010010010110,10011100010011101010011110010011,11101010000001010000101010001001,010001101010010001010010101001,100101001110101000010101001010,101010010
0111100001000100001001100,0010101000010111000011001000010,1100001110100001111000100001000,10001100010010100010011100010100,100010101100010101000101110001,1000100011
1101110101011101110111,10001011110010111010101110111,01111100101111101011111010111,11111000000011000000111000001011,00000111100001001100001011100001,10110000111
110010011100101111001,101110001111110100011110001111,101010111101011110100111101101101101111110000111,1000111110001011110010111110100,11111010101111
1001010000100101100010,011100000100110100010011100001001,111100010100000001010001000101001,000010100110001010100000101010,0010101100010101110001011000000,1011001000101
110100101011100010101,11100010110000001011000100101001,0001011001100101101000010101010,01011011000101011100101110000001,01110010010111010000101101100101,1110000101110
001000110001010001001,001100010011010001010001000101010,10001011001000101110100011000010,00110010100011010010001101101000,11100010001110101000111100100011,111010010000001
011101000011101010,10111010010110111010111000010,11110010101110100101110101011,11100010111101010111110010111,111011000000011000000101000001,001100000101000
101110111011000011,10110010111011010011101101110,1110001110111010111011100111011,11101110000001110000101110001,001111000110111001000111001010,11110011001111001
1010001110110001,1110010001111010001111101000111,1111001000001001000001100100001,0100100001110010010010001011,00100001100100010010001011,001000110101000110111000
1010001110110001,1110010001111010001111101000111,1111001000001001000001100100001,0100100001110010010010010001011,00100001100100010010001011,001000110101000111001000100
0100111101010011,111101010000010101000101010001,010101000111010100100101000101,01010011010101001111010101010101000101,01100110001010110000101111000,011001100001101100001
000001110000001,01100000011110000010011000001011,1000001101100000111100001000110,00010011100001010110000101111000,0110011000011011000011101100001,11111000100001100010
1001011001011,1011001101101100111101101000110,11010011101101010110110111011,011001101101101101101101101,111101110000110111000111011001,0110111001110110110100
1111001000111,100100111100101111001011110,011001111001101111001110111001,11111110100001111010001111101001,0111101001111101010011110101011,111010110111101011111
```

Coordenadas por cantidad de 1's

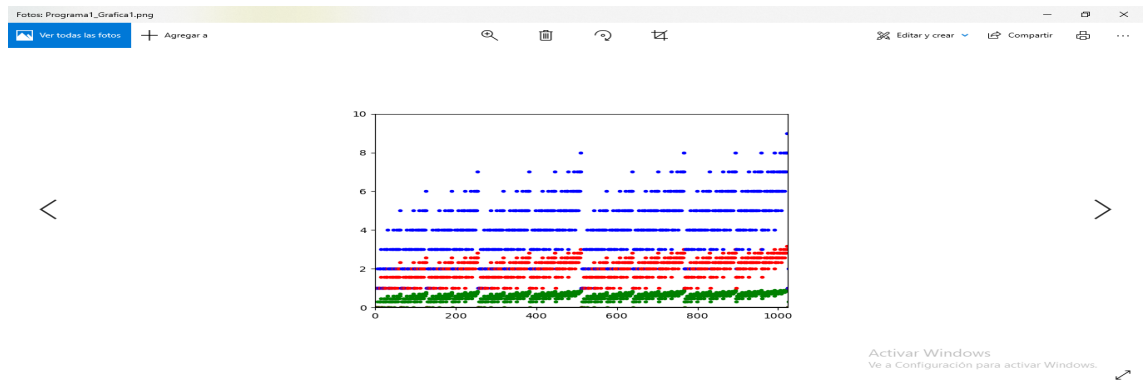
```
Programa1_Coordenadas1: Bloc de notas
Archivo Edición Formato Ver Ayuda
0,0
1,0
2,1
3,0
4,1
5,1
6,2
7,0
8,1
9,1
10,2
11,2
12,2
13,2
14,3
15,0
16,1
17,1
18,2
19,1
20,2
21,2
22,3
23,1
24,2
25,2
26,3
27,2
28,3
29,3
30,4
31,0
32,1
33,1
34,2
35,1
```

Coordenadas por cantidad de 1's (32 bits)

```
Programa1_Coordenadas2: Bloc de notas
Archivo Edición Formato Ver Ayuda
0,14
2,12
3,20
4,9
5,14
6,16
7,18
8,23
9,8
10,10
11,14
12,13
13,15
14,20
15,13
16,16
17,19
18,18
19,21
20,25
21,7
22,9
23,9
24,12
25,12
26,14
27,17
28,11
29,14
30,14
31,16
32,17
33,18
34,22
35,12
36,13
```

Grafica por cantidad de 1's

azul = cantodad de 1's rojo = log2 verde log10



azul = cantidad de 1's rojo = \log_2 verde \log_{10} Grafica por cantidad de 1's (32 bits)

