

Instituto Politécnico Nacional

Escuela Superior de Cómputo



Programa 1: Universo

Autor: Colín Ramiro Joel

Materia: Teoría de la Computación

Grupo: 4CM2

Profesor: Juarez Martínez Gemaro

Fecha de entrega: **12 de Octubre 2021**

Instrucciones

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 y salir hasta que se le especifique.
2. La salida debe ser expresada en notación de conjunto, debe ir a un archivo de texto.
3. Del archivo de salida, graficar el número de 1s de cada cadena. El eje de las "x" representan la cadena y el eje de las "y" el número de 1s que tiene esa cadena. Específicamente en el reporte, calcular y graficar cuando $n=25$.
4. En este reporte debe de estar también el código de la implementación.

Desarrollo

Este Programa se realizó através del lenguaje de programación **Python**. Para comenzzaar, creamos un menú principal, para la selección de funciomamiento del programa. Como preeviamente en la sección de instrucciones se indica, el programa funciona tanto de manera manual(opción 1) como de manera automática(opción 2), además de que tiene una tercera opción la cuál es para fianlizar la ejecución de dicho programa. El usuario mediante la entrada de teclado puede seleccionar cualquiera de las 3 opciones, en el caso de que el usuario digite un caracter no considerado en el menú, le aparece una leyenda que le menciona: *"Opción inválida, Vuelva a intentar"*. Para su realización, recurri al planteamiento de 4 funciones:

1. **permutaciones**.- Recibe 2 argumentos fundamentales, para su funcionamiento que son la cadena y el tamaño de la cadena. La actividad que realiza esta función es que dada una cadena preeviamente establecida como el primer argumento, va a ir permutando los caracteres de la cadena, todo esto hasta que el ciclo llegue al tamaño máximo de la cadena, que esta definido por el segundo argumento.

```

#Función para las permutaciones
def permutaciones(cadena, tam):
    car = list(cadena)
    permutaciones = []

    for i in product(car, repeat = tam):
        permutaciones.append(i)
    return permutaciones

```

2. **calcManual**.- Como su nombre lo implica, esta función realiza el cálculo manual solicitado preeviamente en las instrucciones. Recibe únicamente 1 argumento que es el valor de la n preeviamente definido por el usuario mediante la entrada de teclado. Dentro de esta función se manda a llamar a la función anterior (**permutaciones**) y lo que realiza es que imprime en un archivo txt llamado "SalidaManual.txt". Posterior a su escritura en el archivo txt, se abre nuevamente el archivo para su lectura y para el conteo de todas las ocurrencias del caracter '1'. Esto con la finalidad de observar la cantidad de 1s en la cadena final y proceder a su graficación, esta graficación se lleva a cabo mediante la llamada de la función **grafica**, que posteriormente se va a definir.

```

#Función del Cálculo Manual
def calcManual(n):
    dig = '01'
    tam = n
    aux = tam
    archivo = open('SalidaManual.txt','w')
    archivo.write("{}")
    while aux >= 0:
        archivo.write(str(permutaciones(dig, tam-aux)))
        aux -= 1
    archivo.write("{}")
    print("Cadena escrita en el archivo txt, con exito")
    archivo.close()
    archivo = open('SalidaManual.txt','r')
    cad = archivo.read()
    caract = list(cad)
    unos = caract.count('1') #Eje Y
    ceros = caract.count('0')
    tot = unos + ceros #Eje X
    print("La cantidad de 1s en el archivo es de: " + str(unos))
    archivo.close()
    grafica(unos, tot)

```

3. **calcAuto.**- Para esta tercer función es básicamente similar a la anterior, con la diferencia de que no recibe ningún argumento, ya que en este caso la "n" se calcula por si sola, este cálculo se realiza mediante una función de la libreria "random" (Se debe importar esta libreria), la cual es randint, esta función recibe dos argumentos, los cuales son el rango que puede tener n, algo asi como su dominio, para el caso de esta práctica lo definimos como: **randint(1,1000)**. Además en este caso la cadena preeviamente generada por el programa, se guardará en el archivo "SalidaAuto.txt", calculando los 1s de la cadena como en la función anterior.

```

#Cálculo Automático
def calcAuto():
    n = rd.randint(1,1000)
    print("El valor de n es: " + str(n))
    dig = '01'
    tam = n
    aux = tam
    archivo = open('SalidaAuto.txt','a')
    archivo.write("{")
    while aux >= 0:
        archivo.write(str(permutaciones(dig, tam-aux)))
        aux -= 1
    archivo.write("}")
    print("Cadena escrita en el archivo txt, con éxito")
    archivo.close()
    archivo = open('SalidaAuto.txt','r')
    cad = archivo.read()
    caract = list(cad)
    unos = caract.count('1')
    ceros = caract.count('0')
    tot = unos + ceros #Eje X
    print("La cantidad de 1s en el archivo es de: " + str(unos))
    archivo.close()
    grafica(unos, tot)

```

4. **grafica.**- Finalmente, esta función lo que realiza es la graficación de la cantidad de 1s en la cadena seleccionada ya sea mediante el cálculo manual o automático. Recibe 2 argumentos que son la cantidad de 1s previamente contados en cada función ya sea **calcManual** ó **calcAuto**.

Estos 1s se contarán respecto al número total de 1s y 0s dentro de la cadena que sería el segundo argumento que recibiría esta función. Todo esto se realizó con la función **count**, para su cálculo correcto.

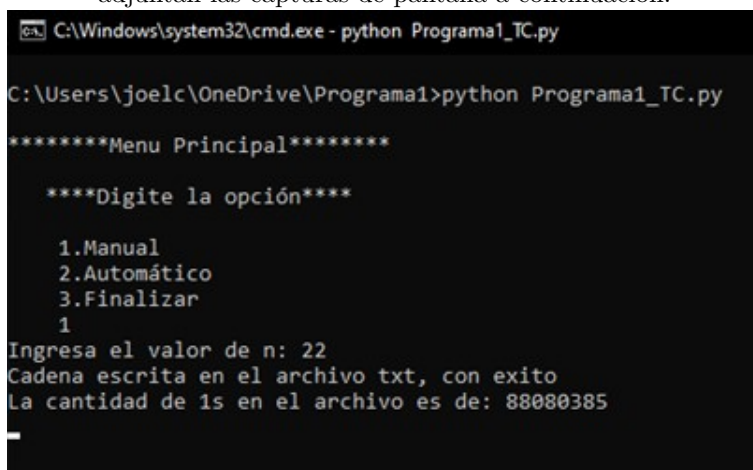
```

#Función que grafica
def grafica(unos, tot):
    x = [0,tot]
    y = [0,unos]
    plt.plot(x,y,color="blue")
    plt.show()

```

Capturas del Funcionamiento

Para comprobar el funcionamiento de este programa, específicamente se tomó en cuenta un $n=25$, en esta sección se encuentran las capturas de pantalla de su funcionamiento, tanto en su cálculo como la gráfica que indica la cantidad de 1s con respecto al total de caracteres. Desafortunadamente, la computadora con la que cuento actualmente no tiene los recursos necesarios para correr el programa con un $n=25$. Intenté correr este programa, en algún servicio de nube pero no fue posible ya que en todas las veces que lo intentaba, me aparecía un error de Memoria. El límite o mejor dicho el número más alto que me permitió correrlo sin problema alguno, fue con un $n = 22$, este proceso se cronometró y tardó alrededor de 5 minutos y medio en completarse, tanto la escritura, como la graficación de la cantidad total de 1s en la cadena. Se adjuntan las capturas de pantalla a continuación.



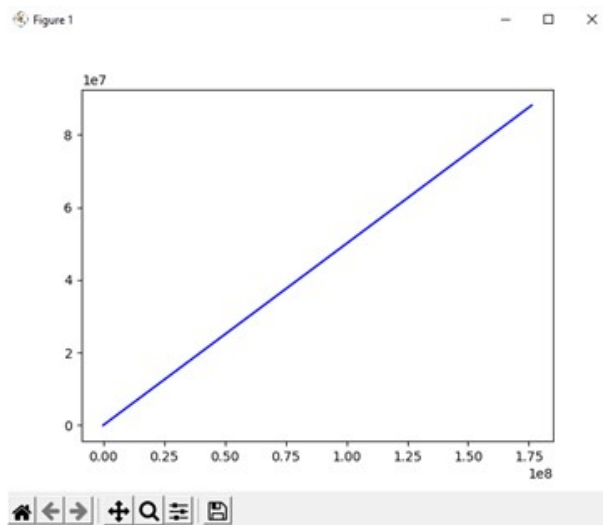
```
C:\Windows\system32\cmd.exe - python Programa1_TC.py

C:\Users\joelc\OneDrive\Programa1>python Programa1_TC.py

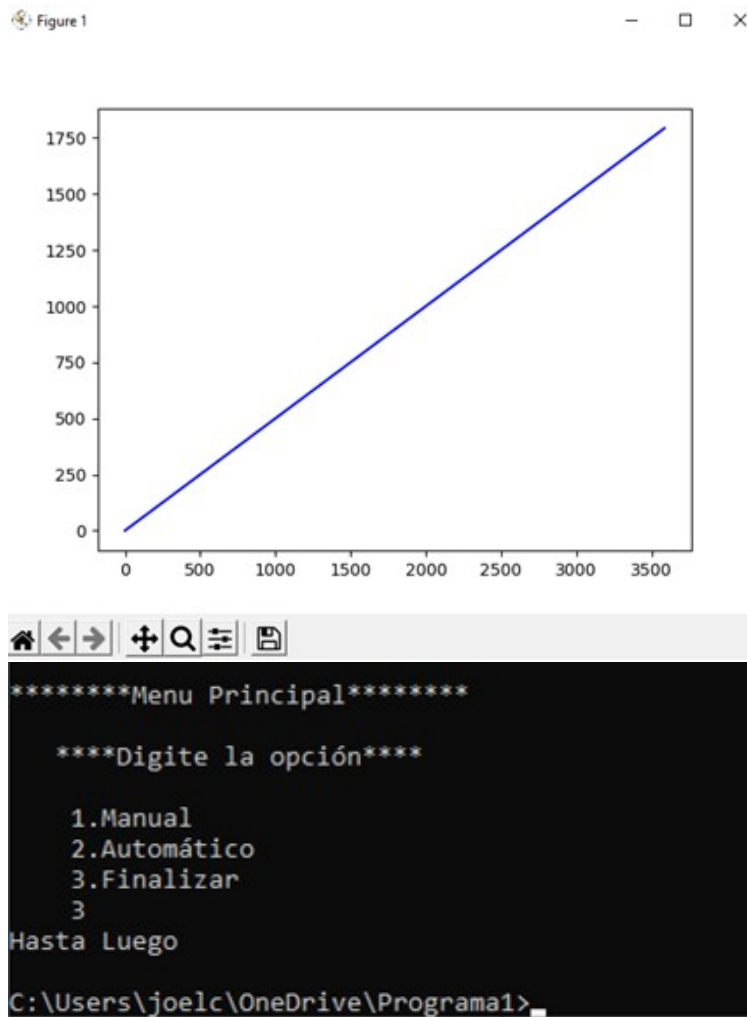
*****Menu Principal*****

    ****Digite la opción****

        1.Manual
        2.Automático
        3.Finalizar
        1
Ingresa el valor de n: 22
Cadena escrita en el archivo txt, con exito
La cantidad de 1s en el archivo es de: 88080385
```



```
*****Menu Principal*****  
  
****Digite la opción****  
  
1.Manual  
2.Automático  
3.Finalizar  
2  
El valor de n es: 8  
Cadena escrita en el archivo txt, con exito  
La cantidad de 1s en el archivo es de: 1793
```



Código

En esta sección se encuentran anexadas capturas del código completo del programa.


```

from itertools import product, repeat, count
import random as rd
import matplotlib.pyplot as plt
#Función para las permutaciones
def permutaciones(cadena, tam):
    car = list(cadena)
    permutaciones = []

    for i in product(car, repeat = tam):
        permutaciones.append(i)
    return permutaciones
#Función del Cálculo Manual
def calcManual(n):
    dig = '01'
    tam = n
    aux = tam
    archivo = open('SalidaManual.txt','w')
    archivo.write("{")
    while aux >= 0:
        archivo.write(str(permutaciones(dig, tam-aux)))
        aux -= 1
    archivo.write("}")
    print("Cadena escrita en el archivo txt, con éxito")
    archivo.close()
    archivo = open('SalidaManual.txt','r')
    cad = archivo.read()
    caract = list(cad)
    unos = caract.count('1') #Eje Y
    ceros = caract.count('0')
    tot = unos + ceros #Eje X
    print("La cantidad de 1s en el archivo es de: " + str(unos))
    archivo.close()
    grafica(unos, tot)

```

```

#Cálculo Automático
def calcAuto():
    n = rd.randint(1,25)
    print("El valor de n es: " + str(n))
    dig = '01'
    tam = n
    aux = tam
    archivo = open('SalidaAuto.txt','w')
    archivo.write("{}")
    while aux >= 0:
        archivo.write(str(permutaciones(dig, tam-aux)))
        aux -= 1
    archivo.write("{}")
    print("Cadena escrita en el archivo txt, con éxito")
    archivo.close()
    archivo = open('SalidaAuto.txt','r')
    cad = archivo.read()
    caract = list(cad)
    unos = caract.count('1')
    ceros = caract.count('0')
    tot = unos + ceros #Eje X
    print("La cantidad de 1s en el archivo es de: " + str(unos))
    archivo.close()
    grafica(unos, tot)
#Función que grafica
def grafica(unos, tot):
    x = [0,tot]
    y = [0,unos]
    plt.plot(x,y,color="blue")
    plt.show()

op = 0
salir = 3
while op != salir:
    print("\n*****Menu Principal*****\n")
    print("    ****Digite la opción****")
    op = int(input('1.Manual\n2.Automático\n3.Finalizar\n'))
    if op == 1:
        n = int(input("Ingresa el valor de n: "))
        calcManual(n)
    elif op == 2:
        calcAuto()
    elif op == 3:
        print("Hasta Luego")
    else:
        print("Opción inválida, Vuelva a intentar")

```