Instituto Politécnico Nacional

Escuela Superior de Computación

Teoría de la Computación

Práctica 2

Lenguaje binario de números primos

Profesor: Juarez Martinez Genaro

Alumno: Jimenez Luna Rodrigo Efren

Email: jimenez.luna.efren@gmail.com

Grupo: 4CM2

Índice

1. Introducción

En el área de las matemáticas, los números primos siempre han sido un tema de gran interés y muchos matemáticos han intentado descubrir sus secretos durante muchos años. Desde sus propiedades, si existe algún límite o si existen infinitos números primos, una forma de crear una función que determine cualquier número primo, etc.

Durante esta práctica abordaremos el tema de los números primos, ya que se creará un programa que genere un determinado número de cadenas que representan los números primos, afrontando los problemas que esto implica.

2. Marco Teórico

2.1. Lenguaje, alfabeto y cadenas

En el campo de la teoría de la computación podemos definir a un alfabeto como un conjunto no vacío el cual contiene símbolos, se utiliza la letra griega ($\sum$) para representarlo, además cuando un símbolo (σ) pertenece a este alfabeto se denotará como $\sigma \epsilon \sum$ [1].

Dentro de un alfabeto encontraremos cadenas, las cuales se componen por una serie de símbolos que pertenecen a un cierto alfabeto. Estas se denotan regularmente por la letra ω [1].

Asimismo, un lenguaje esta formado por las cadenas que genera un determinado alfabeto, este también es un conjunto y se suele representar por la letra L. Añadido a esto, existe la notación de cerradura ($\sum^*$), esta hace referencia al lenguaje que contiene todas las posibles cadenas que se pueden formar a partir de un alfabeto $\sum$ [1].

2.2. Números Primos

Dentro de la teoría de números, los números primos son una parte importante de estudio, esta rama de las matemáticas se ha encargado de analizar las propiedades de estos números y como encontrarlos [2].

La primera noción que se tiene de estos peculiares números se remonta al año 300 a.C. cuando Euclides define a los números primos, demuestra que existen infinitos de ellos y crea el concepto de mínimo y máximo común múltiplo con el famoso algoritmo de Euclides el cual se utiliza para obtener estos últimos. Posteriormente Eratóstenes creo un método sencillo para encontrar números primos, método que recibio el nombre de Criba de Eratóstenes [2].

3. Desarrollo

3.1. Programa 2 Lenguaje binario de números primos

Para esta práctica se plantea realizar un programa que genere un lenguaje binario compuesto por los números primos que se encuentran dentro de un rango n el cuál se encuentra en el intervalo cerrado $[2, 10^7]$.

El programa cuenta con un menú que permita realizar operaciones hasta que el usuario lo indique, la salida se mostrará en notación de conjunto dentro de un archivo de texto. En este caso se crearan dos archivos, el primero con la salida en binario y el otro de forma decimal.

3.2. Planteamiento y solución

Para cumplir los requisitos del problema solo hace falta crear un algoritmo que encuentre los números primos en un intervalo determinado. Por lo que la primera propuesta es probar para cada valor entero dentro del intervalo, probar si este es divisible por algún valor entre 2 y $n-1$. Pero este método resulta bastante ineficiente con una complejidad de tipo $O(N^2)$. Gracias a la teoría de números también sabemos que no es necesario probar todos estos valores hasta $n-1$, ya que solo basta con probar hasta $\sqrt{n}$, resultando en un algoritmo un poco más eficiente.

Como alternativa, es posible implementar el algoritmo de Eratóstenes para obtener números primos, ya que este resulta bastante sencillo de implementar en una computadora. Con la ayuda de este algoritmo, podemos ahorrar muchas operaciones, resultando en un algoritmo más eficiente, por lo tanto, este será el método que se implemento en el código para resolver el problema en cuestión.

3.3. Código

A partir de la solución planteada en la sección anterior, se muestra el código implementado en el lenguaje de programación python:

```
import os
import random
import math
import matplotlib.pyplot as plt

option = 1

while option:
    #Menu
    ##########################################################################
    print("Elija una de las siguientes opciones:")
    print("1) Modo Manual")
```

```
print("2) Modo Aleatorio")
print("3) Salir")

option = int(input())

if option in (1, 2, 3):
    if option == 1:
        print("Indique el valor de n en un intervalo de [2, 10^7]: ")
        size = int(input())

        if size > 10e7 or size < 2:
            os.system("cls")
            print("Valor de n fuera de rango, intente de nuevo!\n")
            continue
    elif option == 2:
        size = random.randint(2, 10**7)
        print("El valor generado para n es {}".format(size))
    else:
        print("\nSesion terminada")
        break
else:
    print("Opcion Invalida")
    continue

print("Espere un momento...")
##############################################################################

# Genera Lenguaje
##############################################################################
binary_file = open("binary_output.txt", "w")
decimal_file = open("decimal_output.txt", "w")

binary_file.write("L = {e")
decimal_file.write("L = {e")

# Algoritmo de Erat stenes para obtener n meros primos
numbers = list(range(2, size + 1))

for p in numbers:
    if p is not None:
        binary_file.write(", " + str(bin(p))[2:])
        decimal_file.write(", " + str(p))
        for multiple in range(p * 2, size + 1, p):
            numbers[multiple - 2] = None

binary_file.write("}"), decimal_file.write("}")
```

```
binary_file.close(), decimal_file.close()
##########################################################################################

with open("binary_output.txt", "r") as file:
    result = file.read()

    ones_count = 0
    y = []

    for char in result:
        if char == "," or char == "}":
            y.append(ones_count)
            # ones_count = 0
        elif char == "1":
            ones_count += 1

plt.plot(y)
plt.xlabel('String')
plt.ylabel('Number of 1s')
plt.title('String and 1s relation')
plt.show()

log10_values = []

for value in y:
    if value != 0:
        log10_values.append(math.log10(value))

plt.plot(log10_values)
plt.xlabel('String')
plt.ylabel('Log10 of 1s')
plt.title('Log10 ones and string relation')
plt.show()

os.system("cls")
```

4. Resultados

Para mostrar que este algoritmo funciona, se realizo una prueba de escritorio haciendo que el programa obtenga los números primos desde 1 hasta 200^3 tal como se muestra en la figura 1.

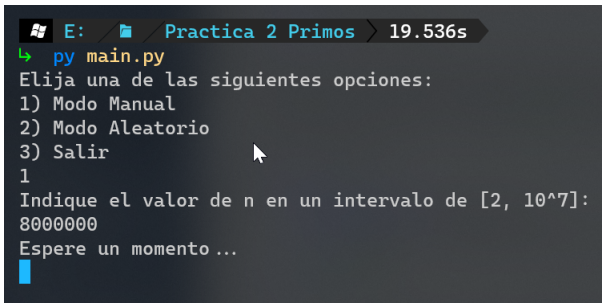

Figura 1. Prueba de escritorio para un valor de $n = 200^3$

Una vez terminado el proceso, observamos la gráfica donde se muestra la cantidad de 1's conforme se añaden cadenas al lenguaje, como se puede observar en la figura 2, esta tiene una forma de línea recta, lo cual tiene sentido ya que la diferencia de 1's entre una cadena y la anterior suele ser constante, a pesar de que no existe un orden con el que los números primos aparecen en los números naturales, al representarlos de forma binaria, estos no difieren tanto en cuanto la cantidad de cifras que se utilizan para representar dichos valores.

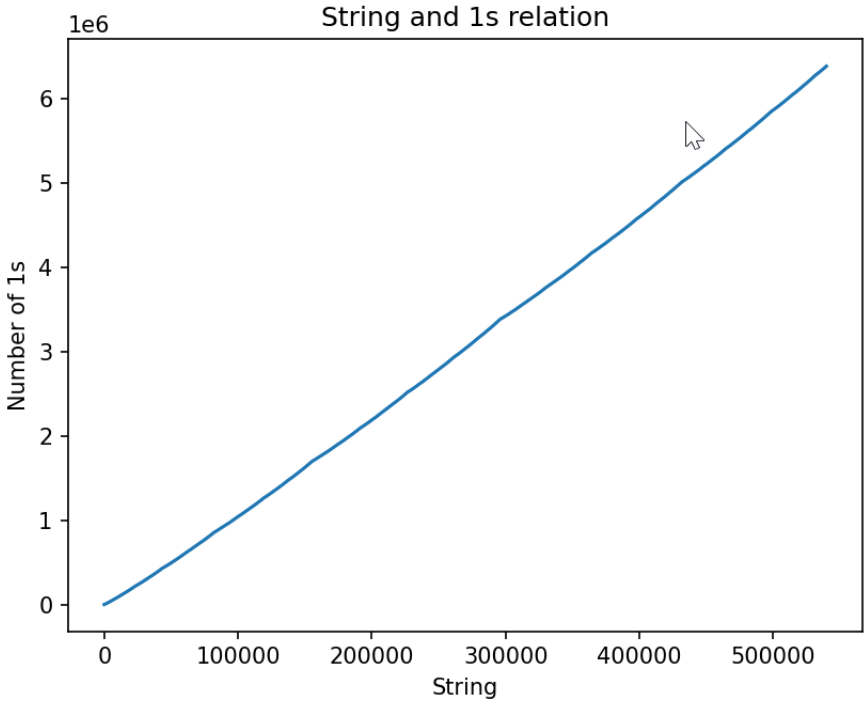

Figura 2. Gráfica de 1's con respecto a las cadenas

Adicionalmente se muestra una segunda gráfica en la figura 3, donde se aplico log_{10} a los valores anteriores mostrando nuevamente que estos tienen una relación lineal haciendo que esta nueva gráfica se comporte de forma logarítmica.

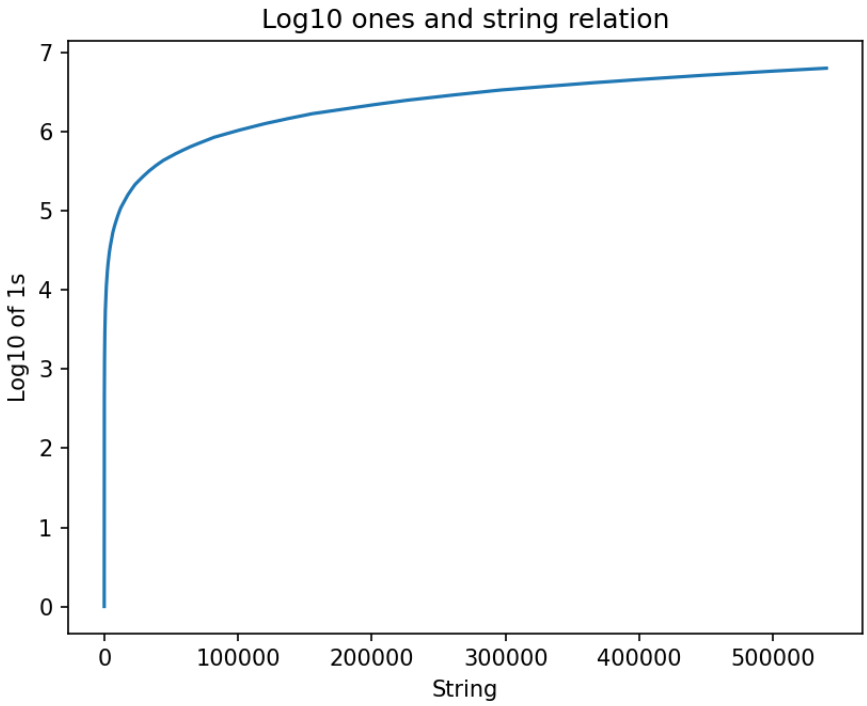

Figura 3. Gráfica aplicando función log_{10} a la cantidad de 1's por cadena.

5. Conclusión

Para la resolución de esta práctica se necesito un conocimiento más profundo de las bases matemáticas que rodean a los números primos los cuales a día de hoy continúan siendo un mundo por explorar y de encontrar nuevas propiedades de estos números.

Gracias al análisis de algoritmos siempre es posible mejorar la eficiencia de un algoritmo, pero en este caso en concreto fue también de mucha utilidad investigar las herramientas que nos proporcionan las matemáticas, al final las computadoras son máquinas muy eficientes para realizar operaciones, por lo que siempre será bueno abarcar los problemas utilizando métodos numéricos.

Referencias

[1] G. Padilla, "Alfabetos, cadenas y lenguajes," 2005. [Online]. Available: http://delta.cs.cinvestav.mx/~mcintosh/comun/summer2006/algebraPablo_html/node4.html

[2] "Número primo," 2023. [Online]. Available: https://es.wikipedia.org/wiki/N%C3%BAmero_primo#Referencias