

# Práctica 2. “Números Primos”

Carmona Serrano Ian Carlo  
Ingeniería en Inteligencia Artificial 5BM1  
Teoría de la Computación , ESCOM- IPN

14 de Octubre 2023

## 1 Introduction

En esta práctica resolveremos el desafío de calcular y mostrar los primeros 10,000,000 de números primos en su forma decimal y binaria. Exploraremos estrategias eficientes para esta tarea, demostrando que es más práctico generar directamente todos los números del universo que verificar su primalidad uno por uno.

## 2 Marco Teórico

Los números primos son números naturales mayores que 1 que solo tienen dos divisores: 1 y ellos mismos. Son elementos fundamentales en teoría de números y tienen diversas aplicaciones en matemáticas y computación.

El algoritmo utilizado en la practica para saber si un número es primo es el siguiente.

1. Descarta números menores o iguales a 1, ya que no pueden ser primos.
2. Acepta el número 2 como primo.
3. Descarta números pares (excepto el 2) porque no son primos.
4. Calcula el límite superior para las pruebas de divisibilidad tomando la raíz cuadrada del número.
5. Verifica si el número es divisible por cualquier número impar desde 3 hasta el límite superior.
6. Si el número es divisible en algún momento, se considera compuesto; de lo contrario, se considera primo.

## 3 Desarrollo

### Paso 1: Selección de Opción

El programa ofrecerá tres opciones al usuario:

1. Ingresar un valor "n" manualmente.
2. Calcular automáticamente un valor "n" dentro del rango  $[2, 10^7]$ .
3. Salir del programa.

```

BIENVENIDO

1) Ingrese la cantidad de primos de forma manual
2) Ingrese la cantidad de primos de forma automatica
3) Salir

Seleccione la opción a realizar: █

```

Figure 1: Paso 1

## Paso 2: Entrada de Datos

Si el usuario elige la opción de ingresar "n" manualmente, se le pedirá que introduzca un valor "n" dentro del rango  $[2, 10^7]$ .

```

Seleccione la opción a realizar: 1
Seleccionaste la opción manual
Ingrese cuantos primos decaea ver: 10000000

```

Figure 2: Paso 2

### Paso 3: Generación de Conjuntos en Forma Binaria y Decimal

A continuación, el programa generará el conjunto  $a$  a partir de los números primos obtenidos:

```

9999347 -> 100110001001001111110011
9999397 -> 100110001001010000100101
9999401 -> 100110001001010000101001
9999419 -> 100110001001010000111011
9999433 -> 100110001001010001001001
9999463 -> 100110001001010001100111
9999469 -> 100110001001010001101101
9999481 -> 100110001001010001111001
9999511 -> 100110001001010010010111
9999533 -> 100110001001010010101101
9999593 -> 100110001001010011101001
9999601 -> 100110001001010011110001
9999637 -> 100110001001010100010101
9999653 -> 100110001001010100100101
9999659 -> 100110001001010100101011
9999667 -> 100110001001010100110011
9999677 -> 100110001001010100111101
9999713 -> 100110001001010101100001
9999739 -> 100110001001010101111011
9999749 -> 100110001001010110000101
9999761 -> 100110001001010110010001
9999823 -> 100110001001010111001111
9999863 -> 100110001001010111110111
9999877 -> 100110001001011000000101
9999883 -> 100110001001011000001011
9999889 -> 100110001001011000010001
9999901 -> 100110001001011000011101
9999907 -> 100110001001011000100011
9999929 -> 100110001001011000111001
9999931 -> 100110001001011000111011
9999937 -> 100110001001011001000001
9999943 -> 100110001001011001000111
9999971 -> 100110001001011001100011
9999973 -> 100110001001011001100101
9999991 -> 100110001001011001110111

```

Figure 3: Paso 3

## Paso 4: Graficación de Número de "Unos"

1. Se calculará el número de unos en cada cadena binaria y se registrará.

2. Se generará una gráfica con el eje x representando las cadenas binarias y el eje y representando el número de unos en cada cadena.
3. Se calculará el logaritmo en base 10 de los valores en el eje y y se generará otra gráfica.

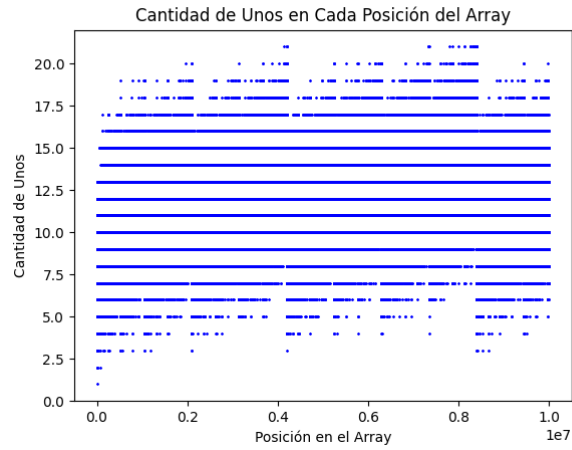


Figure 4: Gráfica de 1's

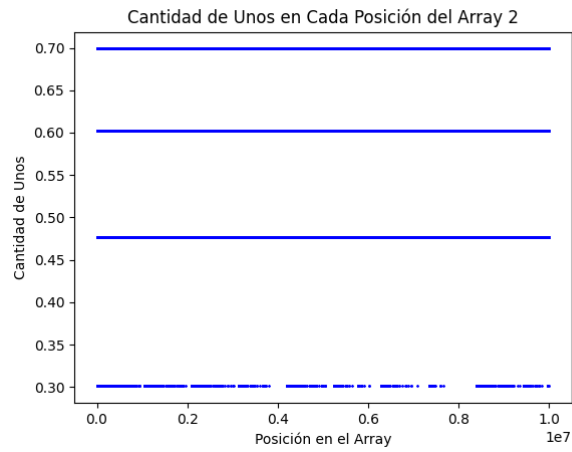


Figure 5: Grafica con log10

## Paso 5: Repetición o Finalización

Después de completar el cálculo y la graficación, el programa preguntará al usuario si desea realizar otra operación.

- El usuario puede elegir ingresar un nuevo valor de "n".
- También tiene la opción de calcular uno automáticamente dentro del rango  $[2, 10^7]$ .
- Por último, puede optar por salir del programa.

## Código

```
1 import math
2 import random
3 import matplotlib.pyplot as plt
4
5 k = 0
6 opcion = 0
7 path = "primos.txt"
8
9
10 def generar_secuencia_binaria(k):
11     if k == 0:
12         return ["0"]
13     elif k == 1:
14         return ["0", "0", "1"]
15     else:
16         nueva_secuencia = ["0", "0", "1"]
17         k_aux = 2
18         k_pos = 0
19
20         while k_aux <= k:
21             for i in range(k_pos + 1, pow(2, k_aux), 1):
22                 nueva_secuencia.append(nueva_secuencia[i] +
23 "0")
24                 nueva_secuencia.append(nueva_secuencia[i] +
25 "1")
26
27             k_pos = pow(2, k_aux) - 1
28             k_aux = k_aux + 1
29
30             del nueva_secuencia[-2:]
31             return nueva_secuencia
32
33 def contar_unos(cadena):
34     contador = 0
```

```

34     for caracter in cadena:
35         if caracter == '1':
36             contador += 1
37
38     return contador
39
40
41 def binario_a_entero(cadena_binaria):
42     try:
43         numero_entero = int(cadena_binaria, 2)
44         return numero_entero
45     except ValueError:
46         return None
47
48
49 def es_primo(numero):
50     if numero <= 1:
51         return False
52     if numero == 2:
53         return True
54     if numero % 2 == 0:
55         return False
56
57     limite_superior = int(math.sqrt(numero)) + 1
58     for i in range(3, limite_superior, 2):
59         if numero % i == 0:
60             return False
61
62     return True
63
64
65 print("BIENVENIDO".center(50))
66 print("\n1)  Ingrese la cantidad de primos de forma manual")
67 print("2)  Ingrese la cantidad de primos de forma automatica
68 ")
69 print("3)  Salir\n")
70
71 opcion = int(input("Seleccione la opci n a realizar: "))
72
73 while opcion != 3:
74     try:
75         if opcion == 1:
76             print("Seleccionaste la opci n manual")
77             k = int(input("Ingrese cuantos primos deca ver:
78 "))
79
80         elif opcion == 2:
81             k = random.randint(0, 10000000)
82             print("Seleccionaste la opci n random")

```

```

82
83     elif opcion == 3:
84         break
85
86     else:
87         print("Opci n no v lida. Por favor, elige 1 o
2.")
88
89     opcion = None
90     bin_k = len(bin(k)) - 2
91
92     print(f"El valor de primos a mostrar -> {k}")
93     input("Presione enter para continuar...")
94
95     secuencia_binaria = list(generar_secuencia_binaria(
bin_k))
96     secuencia_binaria_rango = [palabra for palabra in
secuencia_binaria if len(palabra) == bin_k]
97
98     secuencia_primos = []
99     posiciones = []
100    cantidad_unos = []
101    binario_arreglo = []
102
103    for idx, binario in enumerate(
secuencia_binaria_rango):
104        numero_entero = binario_a_entero(binario)
105        if es_primo(numero_entero) and (numero_entero <=
k):
106            binario_arreglo.append(binario)
107            secuencia_primos.append(numero_entero)
108            posiciones.append(idx)
109            cantidad_unos.append(contar_unos(binario))
110
111    with open(path, "w", encoding="utf-8") as file:
112        for i in range(len(secuencia_primos)):
113            file.write(f"{secuencia_primos[i]} -> {
binario_arreglo[i]}\n")
114        file.close()
115
116    plt.scatter(posiciones, cantidad_unos, marker='*', s
=1, color='blue')
117    plt.xlabel('Posici n en el Array')
118    plt.ylabel('Cantidad de Unos')
119    plt.title('Cantidad de Unos en Cada Posici n del
Array')
120
121    plt.show()
122
123    cantidad_unos_log = []

```

```

124         for i in cantidad_unos:
125             cantidad_unos_log.append(math.log10(
126                 cantidad_unos[i]))
127
128         plt.scatter(posiciones, cantidad_unos_log, marker='*
129         ', s=1, color='blue')
130         plt.xlabel('Posici n en el Array')
131         plt.ylabel('Cantidad de Unos logaritmo 10')
132         plt.title('Cantidad de Unos en Cada Posici n del
133         Array 2')
134
135         plt.show()
136
137         print("\n\n 1)  Ingrese la cantidad de primos de
138         forma manual")
139         print("2)  Ingrese la cantidad de primos de forma
140         automatica")
141         print("3)  Salir\n")
142
143         opcion = int(input("Seleccione la nueva opci n a
144         realizar: "))
145
146     except ValueError:
147         print("Entrada no v lida. Por favor, ingresa un n mero
148         entero.")

```

Listing 1: Practica 2

## 4 Conclusi3n

En esta pr3ctica, hemos abordado el desaf3o de calcular y mostrar los primeros 10,000,000 de n3meros primos en sus formas binaria y decimal. Hemos explorado estrategias eficientes para realizar este c3lculo y hemos demostrado que es m3s pr3ctico generar estos n3meros a partir de una secuencia binaria que verificar su primalidad uno por uno.

Hemos utilizado algoritmos matem3ticos para generar secuencias binarias y convertirlas en n3meros primos. Adem3s, hemos graficado el n3mero de unos en cada cadena binaria y hemos aplicado el logaritmo en base 10 para observar su comportamiento.

## 5 Bibliograf3a

Ullman, J.D. (2009-10). "CS154: Introduction to Automata and Complexity Theory". Sitio web: <http://infolab.stanford.edu/~ullman/ialc/spr10/spr10.html>LECTURE