

Examen Parcial 1

Vega-Lagunas Donaldo F.

Departamento: Computación

Curso: Computación aplicada

Profesor: Benito Granados-Rojas, PhD.

Fecha: 15 de septiembre de 2022

Problema 1 Suma Diagonal Superior: Realizar un programa en lenguaje Python que:

- Solicite al usuario la dimensión m de una matriz, mayor o igual a 3.
- Construya una matriz de $m \times m$ que enumere los primeros $m \times m$ números primos.
- Presente la matriz de manera estructurada (como string, no directamente la lista con corchetes) en pantalla.
- Devuelva la suma de todos los elementos en y por encima de la diagonal principal.
- Probar con diferentes tamaños de matrices.

```
Introduzca la dimensión de la matriz A: 3
La matriz A de números primos consecutivos es:
  2   3   5
  7  11  13
 17  19  23
La suma de los elementos en la matriz diagonal superior es: 57
```

Figura 1: Ejemplo de ejecución de programa en consola de Python

1. Introducción

1.1. Número primo

Un número primo es un número natural mayor que 1 que posee únicamente dos divisores positivos distintos, el mismo y el 1. Al dividirse entre cualquier otro número, se obtiene un residuo distinto de cero.

Los números naturales que tienen un divisor además de sí mismos y del 1, se denominan números compuestos. El número 1 no es considerado primo ni compuesto. Entre los primeros 1000 números naturales existen 168 números primos (Wikipedia s/f(b)).

1.2. Matriz

Una matriz es un arreglo bidimensional de números, se utilizan generalmente para describir sistemas de ecuaciones lineales y registrar datos dependientes de múltiples parámetros.

Una matriz se representa con una letra mayúscula y sus elementos individuales con la correspondiente letra en minúscula y un doble subíndice a_{ij} que indica la fila y la columna de cada elemento, respectivamente. El máximo valor de i es m y el de j es n (Wikipedia s/f(a)).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ a_{m1} & a_{m2} & a_{mn} \end{pmatrix} \quad (1)$$

2. Procedimiento y Resultados

2.1. Identificación de números primos

Para determinar si un número es primo, es necesario definir una función que evalúe cada número generado. Se utiliza `range()` para recorrer todos los números desde 2 hasta $n - 1$, dentro de un ciclo `For` para realizar la evaluación.

La función toma un entero positivo n como argumento, si un número dentro del rango divide

n sin residuo, se concluye que el número evaluado no es primo y devuelve *False*, de lo contrario devuelve *True*.

```
3 #Identificacion de numeros primo
4 def id_primo(n):
5     for i in range(2,n):
6         if(n%i)==0:
7             return False
8     return True
```

Figura 2: Función idprimo en Python

2.2. Generación de números primos

Se define una función para ir almacenando los números primos encontrados por la función de identificación. Se declara una lista vacía y un contador $c = 2$ para omitir el 0 y 1 de la operación. Mientras que la longitud de la lista sea menor al numero de elementos de la matriz definida, se agregan los números primos obtenidos por la función *id_pprimo*, utilizando *list.append* y se incrementa la cuenta del contador.

```
10 #Generacion de lista de n numeros primos
11 def list_primo(n):
12     list=[]
13     c=2
14     while len(list)<n:
15         if id_primo(c):
16             list.append(c)
17         c+=1
18     return list
```

Figura 3: Función listprimo en Python

2.3. Generación de matriz $m \times m$ de números primos consecutivos

Se solicita al usuario el valor de la dimensión m de la matriz deseada, mayor o igual a 3. La condición operativa es que el usuario ingrese un número correspondiente a una dimensión ≥ 3 . Si el número ingresado no cumple con la condición, se notifica al usuario, de lo contrario se ejecuta el bloque de identificación.

La longitud de la lista se obtiene calculando el cuadrado del valor m ingresado por el usuario, y se aplica un formato de arreglo utilizando la instrucción *np.array(nl).reshape(m,m)*.

Para determinar los elementos de la diagonal principal, se utiliza un ciclo for anidado para recorrer cada elemento de la matriz. Cuando el valor de posición de la columna es mayor o igual al de la fila, se realiza la suma del elemento y se almacena en una variable.

```

20 #Generacion de Matriz y suma de elementos en diagonal superior
21 sum=0
22 m=int(input("Introducir dimensión m de una matriz [m >= 3]: "))
23 if m<3:
24     print("La dimensión m debe ser mayor o igual a 3")
25 else:
26     ar=np.array(list_primo(m**2)).reshape(m,m)
27     print("\nMatriz", m, "x", m, "de números primos:\n")

```

Figura 4: Función principal de generación de matriz en Python

En la presentación de la matriz, se debe considerar un espaciado entre elementos variable, que permita visualizar las columnas de manera totalmente vertical. Para esto, es necesario determinar la longitud máxima de los elementos dentro del arreglo, y la longitud actual de cada elemento.

```

28 for row in range(m):
29     line=''
30     for col in range(m):
31         st=str(ar[row][col])
32         if col>=row:
33             sum+=ar[row][col]
34             line+=(str(ar[row][col])+' '*(len(max(st))*4-len(st)))
35     print(line)
36 print("\nSuma de todos los elementos en y por encima de la diagonal principal: ",sum,"\n")

```

Figura 5: Función principal de suma de elementos y presentación en Python

La longitud máxima se multiplica por una constante y se le resta la longitud de cada elemento para mantener la proporción de espacios entre cada elemento, sin importar el número de caracteres que posean.

Por último, se imprime la matriz con el formato correcto y se presenta al usuario la suma total de los elementos sobre la diagonal principal.

```

PS C:\Users\Deathstar\Downloads> python .\E1_1.py
Introducir dimensión m de una matriz [m >= 3]: 3

Matriz 3 x 3 de números primos:

2  3  5
7 11 13
17 19 23

Suma de todos los elementos en y por encima de la diagonal principal: 57

```

Figura 6: Ejecución con dimensión $m=3$ en consola de Python

```

PS C:\Users\Deathstar\Downloads> python .\E1_1.py
Introducir dimensión m de una matriz [m >= 3]: 5

Matriz 5 x 5 de números primos:

2  3  5  7 11
13 17 19 23 29
31 37 41 43 47
53 59 61 67 71
73 79 83 89 97

Suma de todos los elementos en y por encima de la diagonal principal: 482

```

Figura 7: Ejecución con dimensión $m=5$ en consola de Python

```

PS C:\Users\Deathstar\Downloads> python .\E1_1.py
Introducir dimensión  $m$  de una matriz [ $m \geq 3$ ]: 7

Matriz 7 x 7 de números primos:

2  3  5  7  11 13 17
19 23 29 31 37 41 43
47 53 59 61 67 71 73
79 83 89 97 101 103 107
109 113 127 131 137 139 149
151 157 163 167 173 179 181
191 193 197 199 211 223 227

Suma de todos los elementos en y por encima de la diagonal principal: 2013

```

Figura 8: Ejecución con dimensión $m=7$ en consola de Python

3. Conclusiones

Se realizó la identificación de números primos a través de un generador de números naturales, para su uso en una matriz $m \times m$ de longitud deseada por el usuario, con elementos consecutivos. Se utilizó la librería NumPy que permitió pasar los datos almacenados en formato de lista a un arreglo.

La lógica de cálculo para la suma de los elementos superiores a la diagonal principal se obtuvo analizando el orden de los subíndices vistos en la Ec. 1. Se realizaron ejecuciones satisfactorias con dimensiones $m = 3, 5$ y 7 .

Se implementó un espacio variable como solución para la presentación vertical de las columnas de la matriz, sin que se recorrieran los elementos de las filas cuando existía un cambio en la longitud del elemento. Se resolvió utilizando una constante y la diferencia con la longitud actual de los elementos.

4. Repositorio

<https://github.com/a01745982/Examen1>

Referencias

- [1] Wikipedia. *Matriz (matemática)*. s/f. URL: [https://es.wikipedia.org/wiki/Matriz_\(matem%C3%5C%A1tica\)](https://es.wikipedia.org/wiki/Matriz_(matem%C3%5C%A1tica)).
- [2] Wikipedia. *Numero Primo*. s/f. URL: https://es.wikipedia.org/wiki/N%C3%5C%BAmero_primo.