

TRABAJO PRÁCTICO 0

Cuartetos de Números Primos

Los números 11, 13, 17 y 19 tienen una característica interesante. Son 4 números primos y están en la misma decena. El siguiente cuarteto que cumple esa condición es (101, 103, 107 y 109).

Se realizó un programa que calcula y muestra todos los cuartetos de números primos menores que 1.000.000. Este programa es eficaz, pero no eficiente: tarda 31 horas en llegar a un millón.

Se pide diseñar e implementar mejoras al programa para que sea más eficiente. **NO SE DEBE DESARROLLAR UN NUEVO PROGRAMA:** sólo modificar el original.

Requisitos de la Entrega

1. Supuestos: identificar supuestos, limitaciones, condiciones o premisas bajo los cuales funcionará el algoritmo
2. Diseño: Pseudocódigo y estructuras de datos utilizadas
3. Análisis de complejidad: realizar un análisis de complejidad temporal del algoritmo
4. Seguimiento: incluir un ejemplo de seguimiento con los números del 1 al 20
5. Tiempos de Ejecución: medir y graficar los tiempos de ejecución para los números de 1 a 200000, 1 a 400000, 1 a 600000, 1 a 800000 y 1 a 1000000.
6. Informe de Resultados: redactar un informe de resultados comparando los tiempos de ejecución. ¿Se corresponde con la complejidad temporal determinada inicialmente?

Condiciones Generales de Entrega

- El trabajo es individual. Animamos a que discutan alternativas con sus compañeros, pero el desarrollo e implementación de las mejoras y la entrega es individual.
- Si no se entrega, se entrega fuera de plazo, o se entrega algo que no funciona, está desaprobado.
- Los 10 mejores tiempos¹, tienen +1 en la nota final de los TPs
- Si esos tiempos son mejores que el mío (2,46 segundos en un procesador i7 8700 @ 3.2 GHz, 16 GB RAM), se convierte en +2
- Los 5 peores tiempos(*), tienen -1 en la nota final de los TPs
- El trabajo debe ser entregado en un archivo zip conteniendo:
 - Documento con carátula, índice y numeración de páginas. La carátula debe incluir nombre y padrón. Debe presentarse en formato PDF.
 - Archivo(s) con el código fuente del algoritmo desarrollado. Incluir instrucciones para compilar (de ser necesario) y ejecutar.
 - Archivo con los resultados obtenidos
- Si se incluyen referencias bibliográficas, utilizar normas APA 7ma edición

¹ Los tiempos obtenidos en mi computadora serán los únicos válidos para la evaluación

Anexo: Código Fuente (Python)

```
import time

def esprimo(n):

    i = 2
    cd=0
    while i<=n-1:
        if n%i == 0:
            cd = cd+1
            i = i+1
    if cd ==0:
        return True
    else:
        return False

def prog():
    t1=time.time()
    for i in range (11,1000000):
        if esprimo(i) and esprimo(i+2) and esprimo(i+6) and
esprimo(i+8):
            print (i, i+2, i+6,i+8)
    t2=time.time()
    print(t2-t1)

prog()
```