

# TRABAJO PRÁCTICO 0

## **Números Amigos**

Dos números son amigos cuando la suma de los divisores del primero (excepto él mismo) es igual al segundo, y la suma de los divisores del segundo (excepto él mismo) es igual al primero. Por ejemplo, los números 220 y 284 son amigos:

- Los divisores de 220 son 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, y su suma es 284
- Los divisores de 284 son 1, 2, 4, 71 y 142, que suman 220

Se realizó un programa que calcula y muestra todos los pares de números amigos entre 1 y 100.000. Este programa es eficaz, pero no eficiente: tarda 581 segundos (poco menos de 10 minutos) en llegar a 100.000 y más de 18 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 NI CAMBIAR EL PROTOTIPO DE LA FUNCIÓN: sólo modificar el original.

## **Requisitos de la Entrega**

- Supuestos: identificar supuestos, limitaciones, condiciones o premisas bajo los cuales funcionará el algoritmo
- Diseño: Pseudocódigo y estructuras de datos utilizadas
- Análisis de complejidad: realizar un análisis de complejidad temporal del algoritmo
- Seguimiento: Ejemplo de seguimiento con los números del 215 al 225
- Tiempos de Ejecución: medir y graficar los tiempos de ejecución para los números de 1 a 50000, 1 a 100000, 1 a 150000, 1 a 200000 y 1 a 250000.
- Informe de Resultados: redactar un informe de resultados comparando los tiempos de ejecución. ¿Se corresponde con la complejidad temporal determinada inicialmente?
- Alternativas. Explorar fórmulas alternativas para encontrar números amigos

## **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.
- Fecha límite de entrega: 31 de agosto a las 23:59 por campus
- Si no se entrega, se entrega fuera de plazo, o se entrega algo que no funciona, está desaprobado.
- Debe correr en Python 3.10
- La reingeniería debe ser transparente: No cambiar el prototipo de la función
- Los 10 mejores tiempos(\*), tienen +1 en la nota final de los TPs (\*\*).
- 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.
  - Archivos con el código fuente del algoritmo desarrollado. Incluir instrucciones para compilar (de ser necesario) y ejecutar.

- Archivo con los resultados obtenidos
- Aquí no toleramos el plagio: Por ello, se pueden incluir citas en el texto del informe siguiendo el modelo propuesto por Rivas (2022) y luego incorporar el listado completo en un anexo al final, usando normas APA 7ma edición.

(\*) Los tiempos obtenidos en mi computadora serán los únicos válidos para la evaluación

(\*\*) No sirve para convertir un TP desaprobado en aprobado. La nota final de TP no puede ser mayor a 10

### **Anexo: Código Fuente (Python)**

```
import time
def amigos(MAX):

    t1=time.time()
    for i in range(MAX):
        s=0
        for j in range (1,i-1+1):
            if i%j==0:
                s+=j
        s2=0
        for k in range (1,s-1):
            if s%k==0:
                s2+=k
        if i==s2:
            print(i,s)

    t2=time.time()
    print(t2-t1)

amigos(100000)
```

### **Referencias**

- Rivas, A. (2022). Cómo hacer una lista de referencias con Normas APA. Guía Normas APA. <https://normasapa.in/como-hacer-la-lista-de-referencias/>