**Analisis de algoritmo**

* Números perfectos (Hacer un algoritmo)

Para esta prueba se utilizó un procesador I7-9700 3.00GHz con 16 RAM, windows 10 pro

| Valor | Tiempo 1 | Tiempo 2 | Tiempo 3 | Tiempo 4 | Tiempos 5 | Promedio |
| --- | --- | --- | --- | --- | --- | --- |
| 300 | 3456.25 ms | 2681.15ms | 3116.20ms | 4884.89ms | 3613.08ms | 3590.314 |
| 325 | 2531.26ms | 3851.95ms | 3191.03ms | 6414.82ms | 3782.52ms | 3954.316 |
| 400 | 3415.89ms | 3951.98ms | 2732.92ms | 3407.84ms | 3193.86ms | 3340.298 |
| 455 | 2841.46ms | 4407.97ms | 3148.20ms | 2626.92ms | 3491.08ms | 3483.126 |
| 525 | 3195.51ms | 9376.43ms | 2364.90ms | 2742.82ms | 2830.07ms | 4141.87 |
| 800 | 2655.21ms | 2866.39ms | 3606.33ms | 3647.12ms | 3846.73ms | 3344.36 |
| 1000 | 2616.48ms | 3536.77ms | 5837.05ms | 2795.25ms | 3507.51ms | 3758.41 |
| 1025 | 5097.80ms | 5941.12ms | 2866.56ms | 5229.78ms | 2564.82ms | 4339.86 |
| 1096 | 3094.00ms | 4340.22ms | 3363.21ms | 4137.72ms | 3730.24ms | 3733.08 |

import time

def es\_numero\_perfecto(n):

if n <= 1:

return False

suma\_divisores = 0

for i in range(1, n):

if n % i == 0:

suma\_divisores += i

return suma\_divisores == n

inicio = time.time()

numero = int(input("Introduce un número: "))

if es\_numero\_perfecto(numero):

print(f"{numero} El número es perfecto.")

else:

print(f"{numero} No es un número perfecto.")

fin = time.time()

tiempo\_transcurrido = (fin - inicio) \* 1000

print(f"Tiempo de ejecución: {tiempo\_transcurrido:.2f} en milisegundos")

* Para la segunda prueba se utilizó un procesador I7-11700 2.50GHz con 16 RAM, windows 10 pro

| Valor | Tiempo 1 | Tiempo 2 | Tiempo 3 | Tiempo 4 | Tiempos 5 | Promedio |
| --- | --- | --- | --- | --- | --- | --- |
| 300 | 2487.87 | 2231.68 | 3309.24 | 1900.55 | 2233.63 |  |
| 325 | 2763.41 | 4173.55 | 2956.99 | 3034.53 | 3465.78 |  |
| 400 | 1950.28 | 2663.45 | 3045.27 | 1728.39 | 2608.39 |  |
| 455 | 2807.59 | 2309.54 | 2405.44 | 3231.06 | 2260.99 |  |
| 525 | 2744.41 | 4114.98 | 2747.11 | 3343.23 | 2839.56 |  |
| 800 | 2699.44 | 7379.96 | 2214.73 | 2680.95 | 2199.97 |  |
| 1000 | 2397.56 | 2002.41 | 3189.38 | 5401.80 | 2734.91 |  |
| 1025 | 2545.19 | 3195.90 | 3438.17 | 2510.53 | 4430.43 |  |
| 1096 | 3263.40 | 4393.36 | 3536.52 | 4957.52 | 46.10.40 |  |