

Laboratorio práctico 07 – H0622

Experiencia 1

Desarrolle en un cuaderno .ipynb de forma local :

- a) Implemente la siguiente función en Python:

Dado un arreglo de enteros `nums` y un entero `target`, devuelve los índices de los dos números de tal manera que sumen `target`.

Ejemplo:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explicación: nums[0] + nums[1] == 9, se retorna [0, 1]

Input: nums = [3,2,4], target = 6
Output: [1,2]

Input: nums = [3,3], target = 6
Output: [0,1]
```

- b) Llame a la función y para probar con los `nums` y `target` de ejemplo, verificar que funciona correctamente.
- c) Cree una lista aleatoria de números enteros de tamaño `n` y genere un `target` entero aleatorio
- d) Realice una función que llame a la función en el ítem a de forma iterativa y tenga como entrada una lista de tamaño `n`: [50, 100, 200]
- e) Realice un programa iterativo que llame a la función implementada en el ítem d) y realice 100 iteraciones de lectura
- f) Mida el tiempo por cada iteración y genere la gráfica
- g) Suba el cuaderno a Google Colab y ejecute todo su trabajo hasta el momento, genere una gráfica de tiempo e iteración en Google Colab
- h) ¿Los tiempos de ejecución son estables? ¿Qué sugiere realizar si quisiera realizar comparaciones? Comente los resultados obtenidos por su computadora y Google Colab, ¿Cuál es mejor?

Experiencia 2

```
import numpy as np

def benchmark_float_operations(size=1000000):
    a = np.random.rand(size)
    b = np.random.rand(size)
    return a * b
```

```
import os

def benchmark_disk_io():
    data = 'a' * 1024 * 1024 # 1 MB de datos
    with open('test_file.txt', 'w') as f:
        for _ in range(10): # Escribe 10 MB de datos
            f.write(data)
    os.remove('test_file.txt')
```

- a) Implementar variaciones de `benchmark_float_operations` con diferentes tamaños de datos.
 - i. Modifica la función para que acepte un parámetro de tamaño variable.
 - ii. Utiliza módulos como `time` para medir el tiempo de ejecución.
 - iii. Grafica los resultados para visualizar cómo los cambios en el tamaño afectan el rendimiento.
- b) Variaciones de `benchmark_disk_io` con diferentes tamaños de datos:
 - i. Adapta la función para variar el tamaño de los bloques de datos escritos y leídos.
- c) Crea un script en Python que ejecute todas las funciones de benchmarking una tras otra.
- d) Desarrollar un sistema de puntuación basado en las métricas de rendimiento de cada benchmark.
- e) Suba el cuaderno a GoogleColab y ejecute los benchmarks,
- f) Compara los resultados obtenidos en Google Colab con los de otras computadoras.
Realice comentarios sobre los resultados obtenidos

Experiencia 3

- a) Instalar la siguiente librería:
`pip install cryptography`

- b) Correr el script y vea las salidas de las funciones
- c) Cree un programa que mida el tiempo de ejecución que toma cifrar y descifrar datos usando cada uno de los algoritmos. Registre los tiempo de ejecución
- d) Cree archivos de de texto de 1KG, 10KG, 100KB y 1MB
- e) Mida el tiempo de ejecución de cada archivo
- f) Suba sus scripts a Google Colab, instale las librerías necesarias y corra el programa.
- g) Mida el tiempo de ejecución de cada archivo
- h) ¿Qué diferencias encuentra entre los 2 tipos de encriptación? Realice comentarios en base a los resultados de los tiempos de ejecución.