

Estructuras de Datos

Propuesta de Proyecto

Integrantes:

- Juan Diego Vaca
- Christian Santamaria
- Johana Duchi
- Edwin Jaramillo

Tema: Unit testing: Evaluación de algoritmos y tiempos de ejecución.

Nuestro proyecto tiene como objetivo evaluar la correctitud y eficiencia de propuestas de algoritmos como resolución de los problemas planteados en los deberes enviados a lo largo del presente curso, por ello, se han identificado dos componentes principales:

Elaborar un programa que reciba una función (considerando un conocimiento previo de los parámetros de entrada y qué debería retornar), se le aplique una determinada cantidad de pruebas, sencillas y robustas, a fin de determinar si la implementación ha sido correcta, considerando todos los posibles casos.

En esta misma línea de pensamiento, se desea determinar los tiempos de ejecución de los algoritmos evaluados, sus resultados, la estructura de datos utilizada y guardarlos con la finalidad de presentar al usuario las pruebas que se han realizado y, con esa información, evalúe qué algoritmo resulta mejor.

Herramientas necesarias:

1. Para el desarrollo del proyecto se necesitará aplicar *Unit Testing*, un método de chequeo de software, en el que unidades individuales de código fuente son evaluados para determinar si están aptos para ser utilizados. El objetivo es diseñar e implementar los criterios/resultados que son útiles para verificar si la unidad de código puesta a prueba es correcta. Cualquier fallo en el algoritmo evaluado será reportado dentro de un resumen al final de la inspección. Para esto utilizaremos la librería de Python *unittest*, la cual brinda las herramientas necesarias para una correcta implementación de nuestras ideas.
2. Otro punto importante es cómo “importar” y pasar como argumento los algoritmos a evaluar. Existen varias formas de hacerlo, sin embargo, una de las más sencillas es buscar y ejecutar un script de Python que contenga únicamente la función que se quiere testear. Una librería que ayuda en buena manera es *runpy*, ya que cuenta con funciones de localización y ejecución módulos de Python.
3. Dado que se busca tener un registro de las pruebas realizadas, dividiremos los tests de acuerdo al nombre del algoritmo (esto es posible ya que de antemano se conocen los nombres de las funciones), en estas divisiones buscaremos ordenar los registros de la forma más óptima posible dentro de un fichero de texto. Esta última parte se decidirá durante la elaboración del proyecto, debemos comparar la eficiencia del uso y ordenamiento de los registros en los diferentes tipos de datos aprendidos en la clase; nuestras conclusiones se verán reflejadas en el programa final y se expondrá su debida justificación.

Se pretende que, al finalizar este proyecto, se tenga una idea clara de cómo funcionan los Unit Tests, utilizados en diferentes empresas para la elaboración conjunta de software, qué estructuras de datos tienen un mejor performance para determinadas tareas, qué casos pueden mejorar o empeorar el performance de un algoritmo en base a la estructura utilizada, y, finalmente, qué estructura de datos funciona mejor para guardar registros según criterios previamente definidos.

Referencias

Pozo, L. (2019). How to Run Your Python Scripts. <https://realpython.com/run-python-scripts/#:~:text=To%20run%20Python%20scripts%20with,see%20the%20phrase%20Hello%20World!>

Python. (s.f). runpy — Localización y ejecución de módulos Python. <https://docs.python.org/es/3/library/runpy.html>

Sharma, A. (2020). Unit Testing in Python Tutorial. <https://www.datacamp.com/tutorial/unit-testing-python>

How to save and load objects to and from file in Python via facilities from the pickle module. (2018, 4 septiembre). Techcoil Blog. <https://www.techcoil.com/blog/how-to-save-and-load-objects-to-and-from-file-in-python-via-facilities-from-the-pickle-module/>