Brayan Nicolas Daza

Observaciones: Laboratorio 7

i. ¿Por qué en la función getTime() se utiliza time.perf_counter() en vez de otras funciones como time.process_time()?

R/ Consume menor energía, retorna el tiempo en segundos o el derivado que mejor se ubique a la necesidad (milisegundos, microsegundos, nanosegundos, etc), es system-wide y además dicta también los segundos mientras el codigo esta en sueño.

ii. ¿Por qué son importantes las funciones start() y stop() de la librería tracemalloc?

R/ Permiten controlar el rastreo de asignación de memoria en el programa y así evitar un memory leak, o riego de memoria.

iii. ¿Por qué no se puede medir paralelamente el uso de memoria y el tiempo de ejecución de las operaciones?

R/ Mientras el tiempo de ejecución de las operaciones es altamente dependiente del tamaño de la data a procesar, el uso de memoria no depende tanto y puede resultar en una complejidad distinta a la del tiempo.

Por ejemplo, el tiempo para encontrar un valor máximo es O(1)

Mientras que el uso de memoria es O(N).

iv. Teniendo en cuenta cada uno de los requerimientos del reto ¿Cuántos índices implementaría en el Reto? y ¿Por qué?

Uno por cada requerimiento, pues sería ideal para encontrar los elementos de una lista (puede ser series de TV, países, año de publicación, etc) con un menor gasto de memoria y tiempo, además de que permitirse editar para solamente encontrar los primeros tres y últimos tres elementos de una lista.

v. Según los índices propuestos ¿en qué caso usaría Linear Probing o Separate Chaining en estos índices? y ¿Por qué?

En un caso personal, Linear Probing permite no crear nuevas alocaciones a la hora de hacer una inserción (a menos de que se haga Re-Hash) por lo que es ideal para programar en caso de tener baja memoria, por lo que sería lo ideal para trabajar en la mayoría de los índices. Sin embargo, Separate Chaining presenta una menor

posibilidad de degradación en la calidad de la tabla (por causa de una mala función) y no sufre cambios de velocidad en caso de que el factor de carga llegue a 1, SC seria ideal para construir los requisitos con mayor cantidad de información requerida (Es decir, los requisitos grupales.)

vi. Dado el número de elementos de los archivos del reto (large), ¿Cuál sería el factor de carga para estos índices según su mecanismo de colisión?

r/ Idealmente, 1.

Pero en caso de que ocurra una colisión o existan buckets vacíos y demás dificultades, el factor de carga puede llegar hasta 90.