

Preguntas laboratorio 7

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

R/ Porque esta función permite devolver el tiempo en fracciones de segundos en fracciones cortas y con una mejor resolución.

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

R/ Al activar `start()` se empiezan a registrar las asignaciones de memoria, y el `stop()` finaliza el proceso para medir la memoria.

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

R/ Porque si se realiza una medición de forma paralela no va a ser tan precisa. Cuando se mide paralelamente, no se puede medir el tiempo de memoria de forma detallada, y si se mide desde el tiempo de memoria, sería un tiempo más corto, ya que contaría solo desde que empezó a cargar datos.

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

R/ Se pueden utilizar tres índices, uno para habilidades solicitadas, uno para ofertas de trabajo y uno para tipos de contratación. Elegimos estos ya que son los criterios más comunes a la hora de buscar

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

R/ Se puede usar el Separate Chaining ya que es más eficiente en caso de colisiones, aparte tiene un acceso más rápido para los elementos. Sin embargo, cuando elementos del "bucket" varían más de una vez, sería más apropiado usar chaining.

f. 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/ Podría ser entre 0.7 y 0.8 para que tenga equilibrio a la hora de cargar datos y no consuma tanta memoria.

g. ¿Qué cambios percibe en el tiempo de ejecución al modificar el factor de carga máximo para cargar las ofertas de trabajo?

R/ Todo puede depender de la estructura de los datos. Al aumentar el factor de carga, se aumentaría también el consumo de memoria y disminuiría el factor de carga haría más lento el tiempo de ejecución ya que tendría rehashing más frecuente.

h. ¿Qué cambios percibe en el consumo de memoria al modificar el factor de carga máximo para cargar las ofertas de trabajo?

R/ Cuando se aumenta el factor de carga hasta el máximo aumentará aparte de consumir más energía, permitirá que mayor cantidad de elementos se almacenen en el mismo TAD.

i. ¿Qué cambios percibe en el tiempo de ejecución al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

R/ Cuando se usa un Linear Probing puede acceder a los elementos más eficientemente, más no tiene en cuenta que contará con más colisiones. Al usar Separate Chaining puede demorar más al acceder a los datos, sin embargo, ayuda a distribuir las colisiones.

j. ¿Qué cambios percibe en el consumo de memoria al modificar el esquema de colisiones?, si los percibe, describa las diferencias y argumente su respuesta.

R/ El Separate Chaining usa más memoria, ya que tiene que almacenar más listas enlazadas. El Probing usa menos memoria ya que tiene una tabla única.

k. ¿Qué configuración de ideal ADT Map escogería para el índice de ofertas de trabajo por "id"?, especifique el mecanismo de colisión, el factor de carga y el número inicial de elementos

R/ Usaría un "Separate Chaining", ya que tiene mayor eficiencia para colisiones, un factor de carga de 0.8 para un equilibrio, un tamaño inicial de 7000, todo esto para tener una buena eficiencia.