Base de Datos 2 - Laboratorio 02

Profesor: Heider Sanchez

Sección: 3
Grupo: 2
Integrantes: Jesús Valentin Niño Castañeda , Daniel Ignacio Casquino Paz

Análisis de rendimiento

Sequential File (Python):

Nuestra implementación del Ordered Sequential File utiliza punteros, los cuales son representados con un índice y un valor booleano, el cual es verdadero si el puntero apunta hacia el archivo auxiliar.

Inserción de registros:

Para testear el tiempo de ejecución del Sequential File, se insertaron los 1000 registros del archivo **sales_dataset.csv** en orden aleatorio. El orden aleatorio implica la búsqueda de predecesor en el archivo principal (O(lgn)) y en el archivo auxiliar (O(lgn)). Los tiempos de ejecución fueron los siguientes, luego de 10 pruebas:

1. 28738810300 ns
2. 28459558010ns
3. 27497953891ns
4. 27615149497ns
5. 27819229602ns
6. 28460451364ns
7. 28538961648ns
8. 28021444797ns
9. 28690339803ns
10. 28151875019ns

El tiempo promedio de inserción de los 1000 registros es aproximadamente 28199377393ns.

Búsqueda específica:

Para el test de la búsqueda, y en cada uno de los tests siguientes, se insertó el archivo csv de forma aleatoria también. Se ejecutó la búsqueda de un registro de venta con id = 250:

1. 857830ns
2. 841356ns
3. 829689ns
4. 817023ns
5. 873312ns
6. 856466ns
7. 872509ns
8. 861997ns
9. 907826ns
10. 815762ns

El tiempo promedio de búsqueda es 853377ns

Búsqueda por rango:

La búsqueda por rango se ejecuta con el intervalo [250, 750]. Hace la búsqueda del menor elemento encontrado dentro del intervalo, y utiliza los punteros para añadir el resto de elementos de forma lineal.

1. 979661 ns
2. 15274286 ns
3. 3682851 ns
4. 1012802 ns
5. 1890000 ns
6. 2114000 ns
7. 5655491 ns
8. 1876000 ns
9. 1671500 ns
10. 2045622 ns

Tiempo de ejecución promedio: 3523621 ns

Eliminación de registro:

Para el test de eliminación, se eliminó el registro con ID 250. Se utiliza búsqueda binaria en el archivo principal, y búsqueda lineal en el archivo auxiliar. El tiempo de ejecución fue 0 cuando el archivo no se encontró en los archivos

1. 0ns
2. 1940727 ns
3. 0ns
4. 1000642 ns
5. 0ns
6. 998973 ns
7. 0ns
8. 1435995 ns
9. 995635 ns
10. 0ns

Tiempo de eliminación promedio: 1274594 ns

AVL File (C++):

Inserción de registros:

Se insertar todos los registros presenten en el archivo “sales_dataset_csv”.
Tiempo de las pruebas:

11. 649268800 ns
12. 642018300 ns
13. 679312800 ns
14. 641029800 ns
15. 649933000 ns
16. 651133900 ns
17. 649290800 ns
18. 601474200 ns
19. 629497100 ns
20. 638256800 ns

Promedio: 643,121,550 ns → 643.12 ms

Búsqueda específica:

Se busca por el registro con ID 250.
Tiempo de las pruebas:

11. 591100 ns
12. 574800 ns

13. 1118700 ns
14. 581300 ns
15. 518400 ns
16. 577800 ns
17. 19673100 ns
18. 2369700 ns
19. 601200 ns
20. 912400 ns

Promedio: 2,751,850 ns → 2.75 ms

Búsqueda por rango:

Se buscaron registros en el rango de ID [250, 750].
Tiempo de las pruebas:

11. 1871200 ns
12. 1851300 ns
13. 1892300 ns
14. 2271200 ns
15. 1890000 ns
16. 4362200 ns
17. 2027600 ns
18. 1876000 ns
19. 2043400 ns
20. 1861000 ns

Promedio: 2,205,520 ns → 2.21 ms

Eliminación de registros:

Se eliminaron todos los registros presentes en el archivo.
Tiempo de las pruebas:

11. 516353400 ns
12. 429497100 ns
13. 425371100 ns
14. 439612200 ns
15. 442813500 ns
16. 456942000 ns
17. 457047900 ns
18. 449416700 ns
19. 415445700 ns
20. 406685200 ns

Promedio: 443,818,480 ns → 443.82 ms