

Hoja de trabajo # 6

c) Use un profiler para evaluar el tiempo de ejecución de su programa para mostrar las cartas. Corra su programa con las tres implementaciones y muestre los tiempos de ejecución de cada una de ellas. Diga cuál es la más rápida con el profiler.

Luego de haber realizado la ejecución del programa realizando todas las operaciones solicitadas y con la ayuda del profiler, se pudieron obtener los siguientes resultados:

HashMap:

main					
Main.main ()		117,111 ms	(100%)	101,404 ms	(100%)
Controlador.agregarProducto ()		117,111 ms	(100%)	101,404 ms	(100%)
java.util.Scanner.next ()		51,204 ms	(43.7%)	51,204 ms	(50.5%)
Main.readFile ()		50,200 ms	(42.9%)	50,200 ms	(49.5%)
javax.swing.JFileChooser.showOpenDialog ()		15,706 ms	(13.4%)	0.0 ms	(0%)
Self time		15,706 ms	(13.4%)	0.0 ms	(0%)
Self time		0.0 ms	(0%)	0.0 ms	(0%)
Self time		0.0 ms	(0%)	0.0 ms	(0%)

TreeMap:

Name		Total Time	
main		39,709 ms	(100%)
Main.main ()		39,709 ms	(100%)
java.util.Scanner.next ()		28,101 ms	(70.8%)
Controlador.agregarProducto ()		8,000 ms	(20.1%)
java.util.Scanner.nextLine ()		4,192 ms	(10.6%)
Controlador.verif ()		3,807 ms	(9.6%)
Self time		0.0 ms	(0%)
java.util.Scanner.nextLine ()		3,607 ms	(9.1%)
Self time		0.0 ms	(0%)

LinkedHashMap

En los cuales se puede notar que,

main					
Main.main ()		50,502 ms	(100%)	50,502 ms	(100%)
java.util.Scanner.next ()		50,502 ms	(100%)	50,502 ms	(100%)
Controlador.agregarProducto ()		38,894 ms	(77%)	38,894 ms	(77%)
java.util.Scanner.nextLine ()		8,000 ms	(15.8%)	8,000 ms	(15.8%)
Self time		3,607 ms	(7.1%)	3,607 ms	(7.1%)
Self time		0.0 ms	(0%)	0.0 ms	(0%)

de las tres implementaciones, la más rápida para realizar sus operaciones fue TreeMap. Obteniendo un tiempo menor, de las cuales la más lenta fue el HashMap original. Esto puede ser debido a como TreeMap maneja sus datos y que linkedHashMap es una estructura mas compleja.

d) Calcule la complejidad de tiempo para la implementación HashMap, para mostrar todas las cartas. Indique como llegó a ese resultado.

Como se puede observar, el Hashmap teóricamente tiene una complejidad en sus operaciones y búsquedas de $O(n)$, ya que aunque en muchos de los casos esta puede llegar a ser de $O(1)$, debemos tomar en cuenta que esto no se puede garantizar y tomamos como teórico el valor en el peor de los casos que es $O(n)$.