Universidad del Valle de Guatemala
Algoritmos y estructuras de datos
Carlos Calderón, 15219
Jorge Azmitia, 15201

Análisis

Inciso c

Estas tablas corresponden a las corridas que se hicieron del programa con las 3 implementaciones de la interfaz Set. Como puede observarse, la Tabla 1 y 3 son muy similares, según la teoría esto debería ser así pues ambas tienen un rendimiento constante. Por otra parte, puede verse que a grandes cantidades el TreeSet tiende a ser más lento, debido a que su complejidad es logarítmica. En base a los resultados, puede mencionarse que HashSet es la implementación más rápida. Aunque a pequeña escala las 3 rinden muy parecido.

No. Desarrolladores	Tiempo (s)
5	0.001
50	0.002
500	0.003
5000	0.006
50000	0.034
500000	0.596

Tabla 1. HashSet

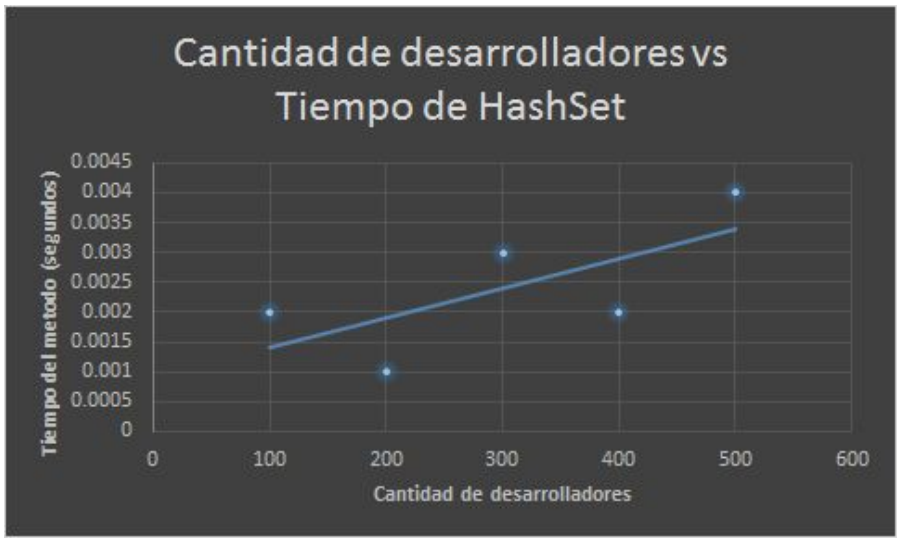
No. Desarrolladores	Tiempo (s)
5	.002
50	.003
500	0.004
5000	0.008
50000	0.062
500000	0.809

Tabla 2. Treeset

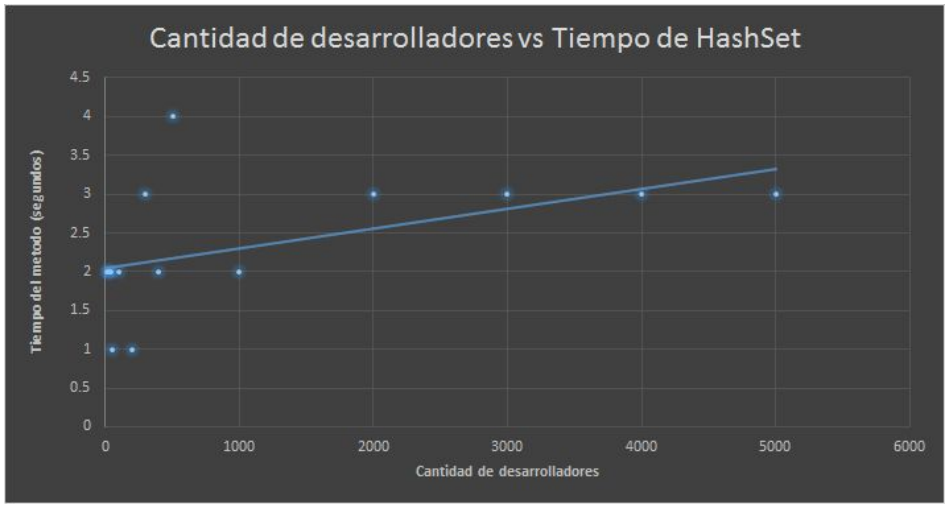
No. Desarrolladores	Tiempo (s)
5	0.001
50	.002
500	.003
5000	0.006
50000	0.036
500000	0.692

Tabla 3. LinkedHashSet

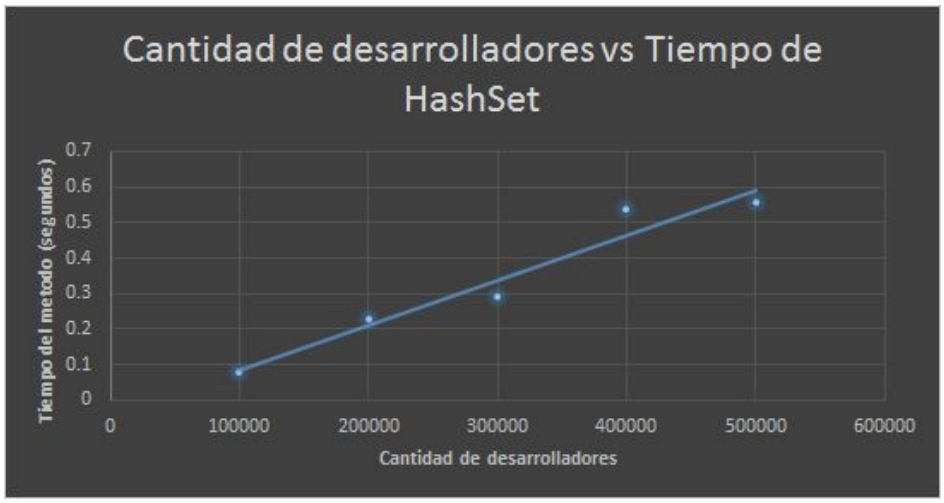
Inciso d.



Gráfica 1. Cantidad de desarrolladores vs Tiempo de Hashset v1.



Gráfica 2. Cantidad de desarrolladores vs Tiempo de HashSet v2.



Gráfica 3. Cantidad de desarrolladores vs Tiempo de HashSet v3.

Para abordar el problema se hicieron 3 gráficas que demostraran la misma información pero con distinto número de desarrolladores. Esto con el fin de hacer notar que a pequeña escala se nota muy bien que el HashSet tiene orden O(1). Como es el caso de la Gráfica 1 y 2. En los que los valores oscilan entre 10 y 5000. Sin embargo, para valores mucho más significativos que es el caso de cientos de miles como se observa en la Gráfica 3. Pareciera ser que la gráfica se comporta de modo distinto, queriendo de alguna forma tender a ser algo diferente al orden ya mencionado. No obstante, la diferencia no es abismal, como para considerar un cambio de comportamiento o de orden. Por lo tanto, se comprueba que la implementación HashSet es de orden O(1).