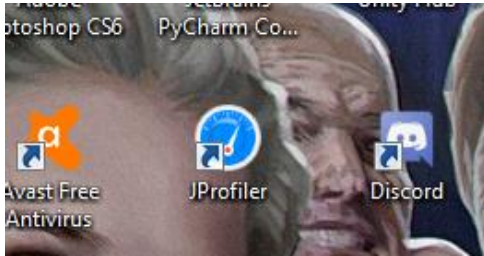


Profiler utilizado: JProfiler (IntelliJ)**HashMap : Mas lento**

- ▼ **m** 100.0% – 30,802 ms – 1 inv. Main.main
 - ▶ **m** 0.1% – 19,064 µs – 1 inv. MostrarElementosProfiler.mostrarElementos
 - ▶ **m** 0.0% – 12,002 µs – 1 inv. java.lang.ClassLoader.loadClass
 - ▶ **m** 0.0% – 600 µs – 0 inv. MostrarElementosProfiler.<init>

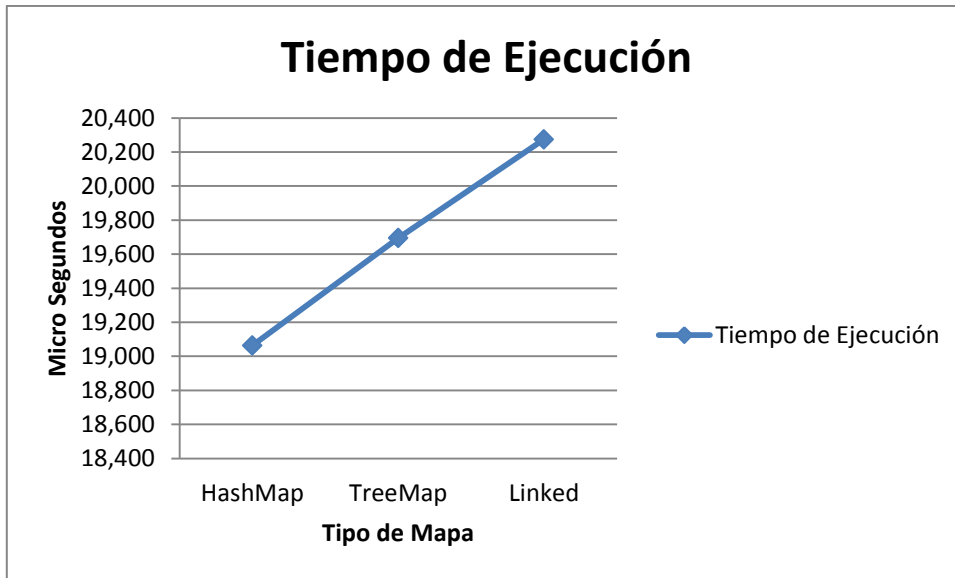
TreeMap: Intermedio

- ▼ **m** 100.0% – 6,605 ms – 1 inv. Main.main
 - ▶ **m** 0.3% – 19,695 µs – 1 inv. MostrarElementosProfiler.mostrarElementos
 - ▶ **m** 0.2% – 15,220 µs – 1 inv. java.lang.ClassLoader.loadClass
 - ▶ **m** 0.0% – 453 µs – 0 inv. MostrarElementosProfiler.<init>

LinkedMap: Mas rápido

- ▼ **m** 100.0% – 7,466 ms – 1 inv. Main.main
 - ▶ **m** 0.3% – 20,275 µs – 1 inv. MostrarElementosProfiler.mostrarElementos
 - ▶ **m** 0.2% – 12,731 µs – 1 inv. java.lang.ClassLoader.loadClass
 - ▶ **m** 0.0% – 424 µs – 0 inv. MostrarElementosProfiler.<init>

Como se puede ver con nuestro profiler, podemos concluir que el mapa más lento es el HashMap, mientras que el más rápido es el LinkedHashMap. A continuación se puede ver una gráfica donde se muestra.



Complejidad de Hashmap

Puedo concluir que la complejidad del HashMap es de $O(n)$. Mi respuesta se basa en la investigación previa en la cual pude descubrir que un HashMap debe de tomar todos los datos que el mapa contiene (n datos). Entonces de la cantidad de datos va a depender este mismo.

Referencias

<https://codeday.me/es/qa/20190107/77546.html>

<https://code.i-harness.com/es/q/457b98>

<https://docs.oracle.com/javase/8/docs/api/java/util/Map.html>