UNIVERSIDAD DEL VALLE DE GUATEMALA

Guatemala 03 de marzo de 2019

Algoritmos y Estructura de Datos

Hoja de Trabajo N°6: Operaciones con MAPs

Pablo Sao - 11530

Según las pruebas expuestas por Eugin (2018) en su explicación, se procedió a realizar una prueba similar con diversas cantidades de datos (cartas) agregadas a las diversas implementaciones del MAP, cargando en la primera prueba todos los tipos de las cartas del listado, dando 8861 registros (cartas), en la segunda prueba se omitieron la cartas con el tipo “Trampa”, cargando 7524 registros (cartas).

Al graficar los datos obtenidos en el profailer (ver sección de profiler de este documento) para ambas pruebas, se pudo corroborar que la operación para mostrar los datos tiene una complejidad O(1), sin embargo para ordenar los datos dentro de un MAP, su complejidad es de O(n). Para cargar datos dentro del MAP, se puede observar que unas poseen un mejor rendimiento, sin embargo su comportamiento es casi contante, por lo que puede llegar a decirse que su complejidad es de O(1).

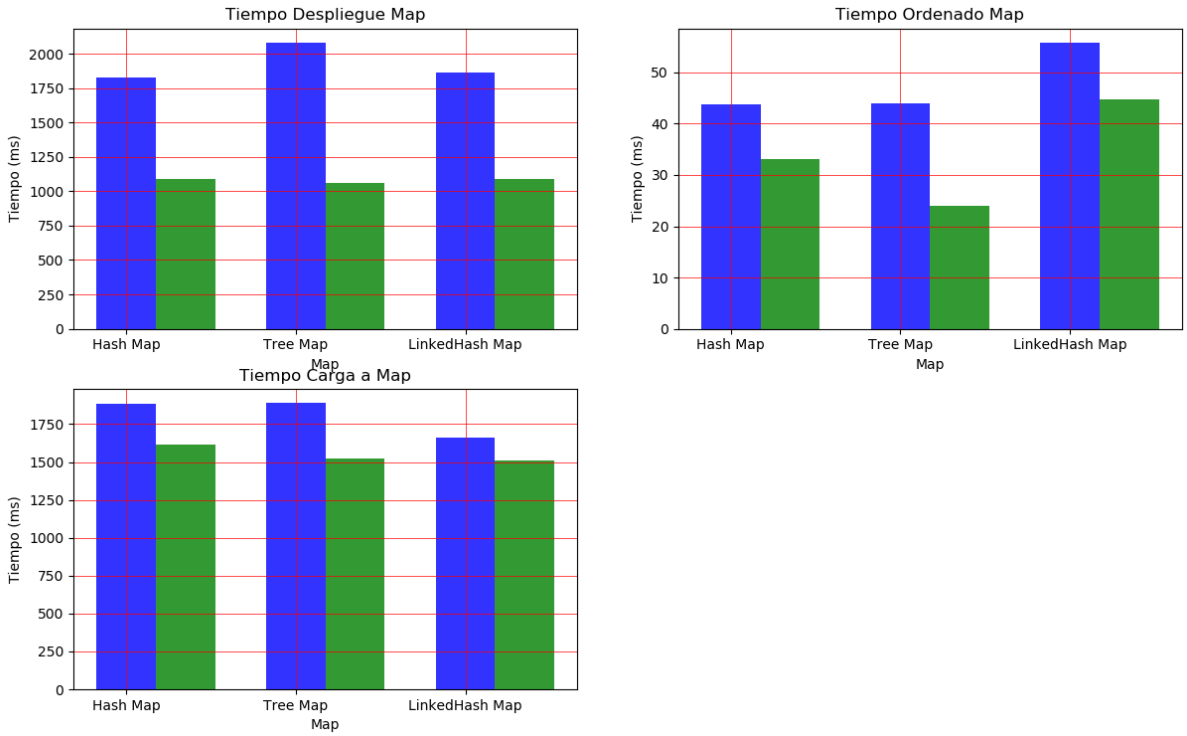
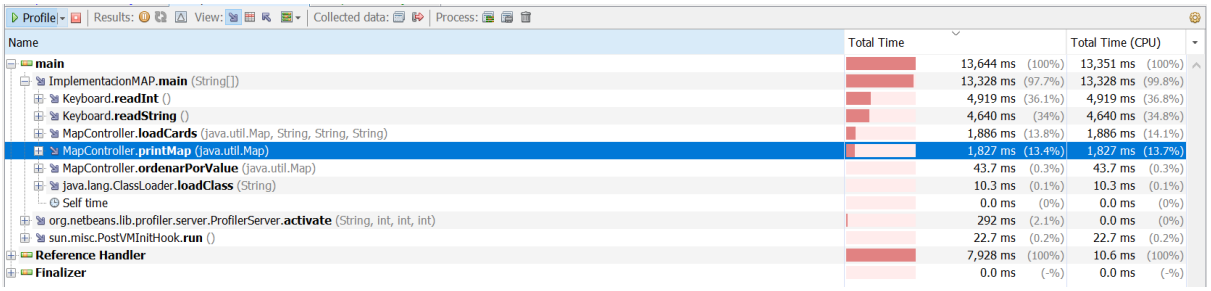


Figura 1: Grafica comparativa de tiempos.

Profiler

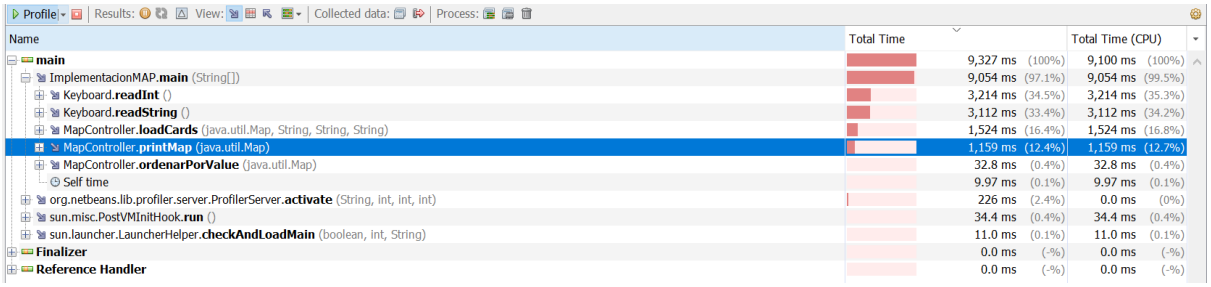
HashMap

Todas la Cartas



Name	Total Time	Total Time (CPU)
main	13,644 ms (100%)	13,351 ms (100%)
ImplementacionMAP.**main** (String[])	13,328 ms (97.7%)	13,328 ms (99.8%)
Keyboard.**readInt** ()	4,919 ms (36.1%)	4,919 ms (36.8%)
Keyboard.**readString** ()	4,640 ms (34%)	4,640 ms (34.8%)
MapController.**loadCards** (java.util.Map, String, String, String)	1,886 ms (13.8%)	1,886 ms (14.1%)
MapController.**printMap** (java.util.Map)	1,827 ms (13.4%)	1,827 ms (13.7%)
MapController.**ordenarPorValue** (java.util.Map)	43.7 ms (0.3%)	43.7 ms (0.3%)
java.lang.ClassLoader.**loadClass** (String)	10.3 ms (0.1%)	10.3 ms (0.1%)
Self time	0.0 ms (0%)	0.0 ms (0%)
org.netbeans.lib.profiler.server.ProfilerServer.**activate** (String, int, int, int)	292 ms (2.1%)	0.0 ms (0%)
sun.misc.PostVMInitHook.**run** ()	22.7 ms (0.2%)	22.7 ms (0.2%)
Reference Handler	7,928 ms (100%)	10.6 ms (100%)
Finalizer	0.0 ms (-%)	0.0 ms (-%)

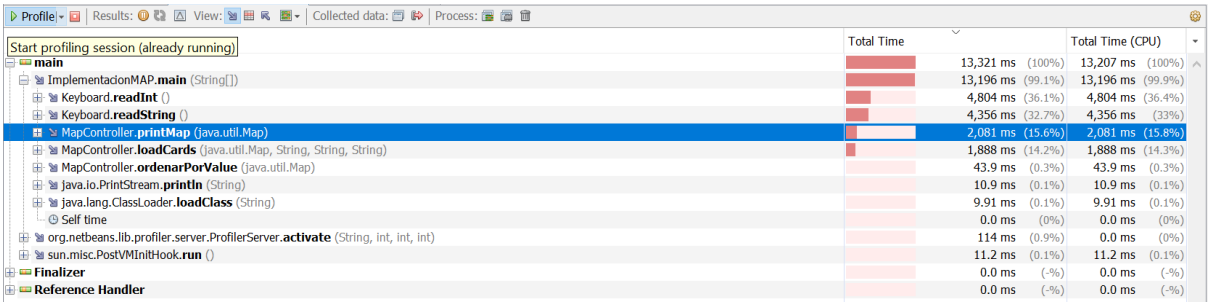
Sin agregar carta “Trampa”



Name	Total Time	Total Time (CPU)
main	9,327 ms (100%)	9,100 ms (100%)
ImplementacionMAP.**main** (String[])	9,054 ms (97.1%)	9,054 ms (99.5%)
Keyboard.**readInt** ()	3,214 ms (34.5%)	3,214 ms (35.3%)
Keyboard.**readString** ()	3,112 ms (33.4%)	3,112 ms (34.2%)
MapController.**loadCards** (java.util.Map, String, String, String)	1,524 ms (16.4%)	1,524 ms (16.8%)
MapController.**printMap** (java.util.Map)	1,159 ms (12.4%)	1,159 ms (12.7%)
MapController.**ordenarPorValue** (java.util.Map)	32.8 ms (0.4%)	32.8 ms (0.4%)
Self time	9.97 ms (0.1%)	9.97 ms (0.1%)
org.netbeans.lib.profiler.server.ProfilerServer.**activate** (String, int, int, int)	226 ms (2.4%)	0.0 ms (0%)
sun.misc.PostVMInitHook.**run** ()	34.4 ms (0.4%)	34.4 ms (0.4%)
sun.launcher.LauncherHelper.**checkAndLoadMain** (boolean, int, String)	11.0 ms (0.1%)	11.0 ms (0.1%)
Finalizer	0.0 ms (-%)	0.0 ms (-%)
Reference Handler	0.0 ms (-%)	0.0 ms (-%)

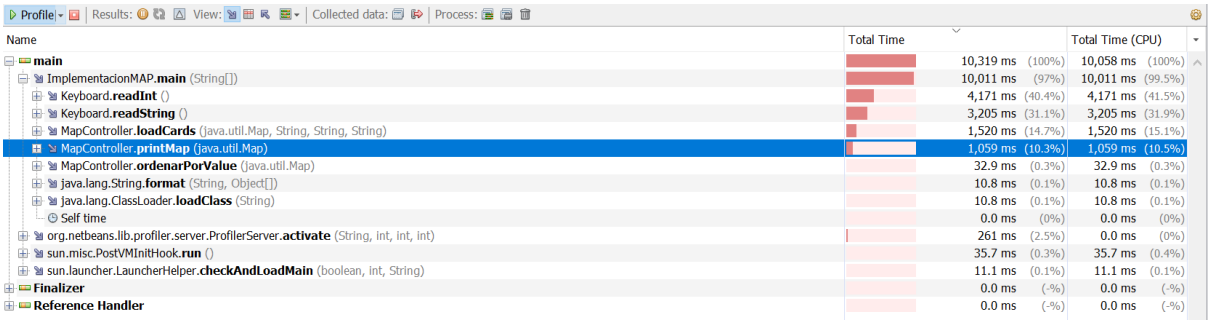
TreeMap

Todas la Cartas



Start profiling session (already running)	Total Time	Total Time (CPU)
main	13,321 ms (100%)	13,207 ms (100%)
ImplementacionMAP.**main** (String[])	13,196 ms (99.1%)	13,196 ms (99.9%)
Keyboard.**readInt** ()	4,804 ms (36.1%)	4,804 ms (36.4%)
Keyboard.**readString** ()	4,356 ms (32.7%)	4,356 ms (33%)
MapController.**printMap** (java.util.Map)	2,081 ms (15.6%)	2,081 ms (15.8%)
MapController.**loadCards** (java.util.Map, String, String, String)	1,888 ms (14.2%)	1,888 ms (14.3%)
MapController.**ordenarPorValue** (java.util.Map)	43.9 ms (0.3%)	43.9 ms (0.3%)
java.io.PrintStream.**println** (String)	10.9 ms (0.1%)	10.9 ms (0.1%)
java.lang.ClassLoader.**loadClass** (String)	9.91 ms (0.1%)	9.91 ms (0.1%)
Self time	0.0 ms (0%)	0.0 ms (0%)
org.netbeans.lib.profiler.server.ProfilerServer.**activate** (String, int, int, int)	114 ms (0.9%)	0.0 ms (0%)
sun.misc.PostVMInitHook.**run** ()	11.2 ms (0.1%)	11.2 ms (0.1%)
Finalizer	0.0 ms (-%)	0.0 ms (-%)
Reference Handler	0.0 ms (-%)	0.0 ms (-%)

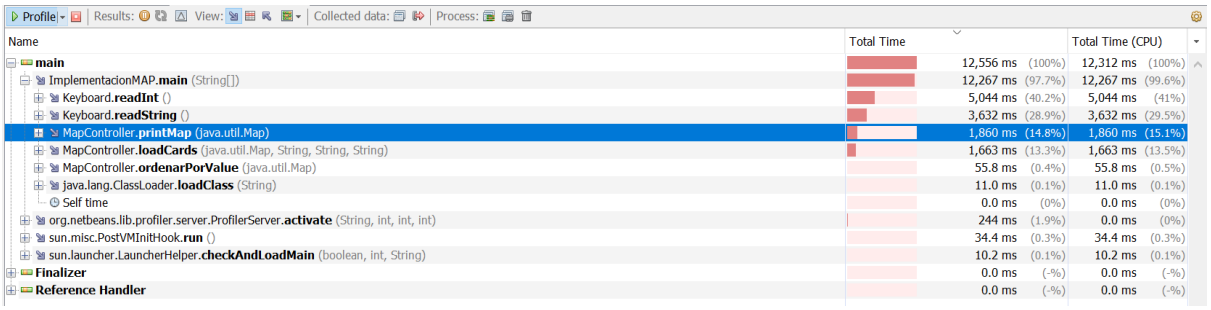
Sin agregar carta “Trampa”



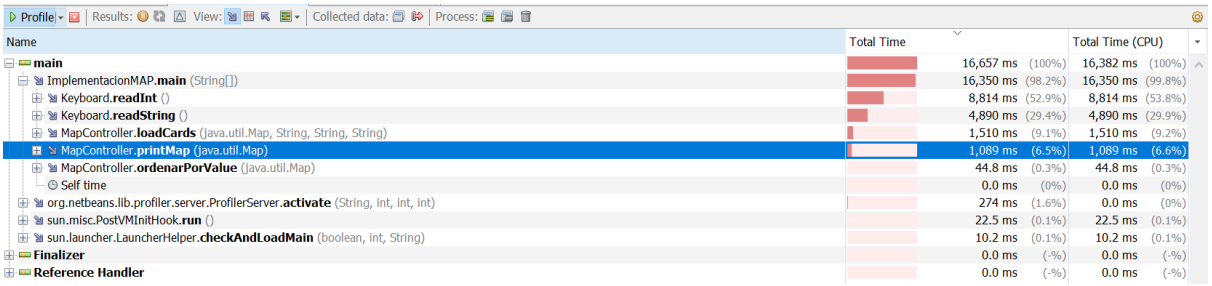
Name	Total Time	Total Time (CPU)
main	10,319 ms (100%)	10,058 ms (100%)
ImplementacionMAP.**main** (String[])	10,011 ms (97%)	10,011 ms (99.5%)
Keyboard.**readInt** ()	4,171 ms (40.4%)	4,171 ms (41.5%)
Keyboard.**readString** ()	3,205 ms (31.1%)	3,205 ms (31.9%)
MapController.**loadCards** (java.util.Map, String, String, String)	1,520 ms (14.7%)	1,520 ms (15.1%)
MapController.**printMap** (java.util.Map)	1,059 ms (10.3%)	1,059 ms (10.5%)
MapController.**ordenarPorValue** (java.util.Map)	32.9 ms (0.3%)	32.9 ms (0.3%)
java.lang.String.**format** (String, Object[])	10.8 ms (0.1%)	10.8 ms (0.1%)
java.lang.ClassLoader.**loadClass** (String)	10.8 ms (0.1%)	10.8 ms (0.1%)
Self time	0.0 ms (0%)	0.0 ms (0%)
org.netbeans.lib.profiler.server.ProfilerServer.**activate** (String, int, int, int)	261 ms (2.5%)	0.0 ms (0%)
sun.misc.PostVMInitHook.**run** ()	35.7 ms (0.3%)	35.7 ms (0.4%)
sun.launcher.LauncherHelper.**checkAndLoadMain** (boolean, int, String)	11.1 ms (0.1%)	11.1 ms (0.1%)
Finalizer	0.0 ms (-%)	0.0 ms (-%)
Reference Handler	0.0 ms (-%)	0.0 ms (-%)

LinkedHash Map

Todas la Cartas



Sin agregar carta “Trampa”



Referencias

Eugin. (2018). *Time Complexity of Java Collections*. Extraído de: https://www.baeldung.com/java-collections-complexity