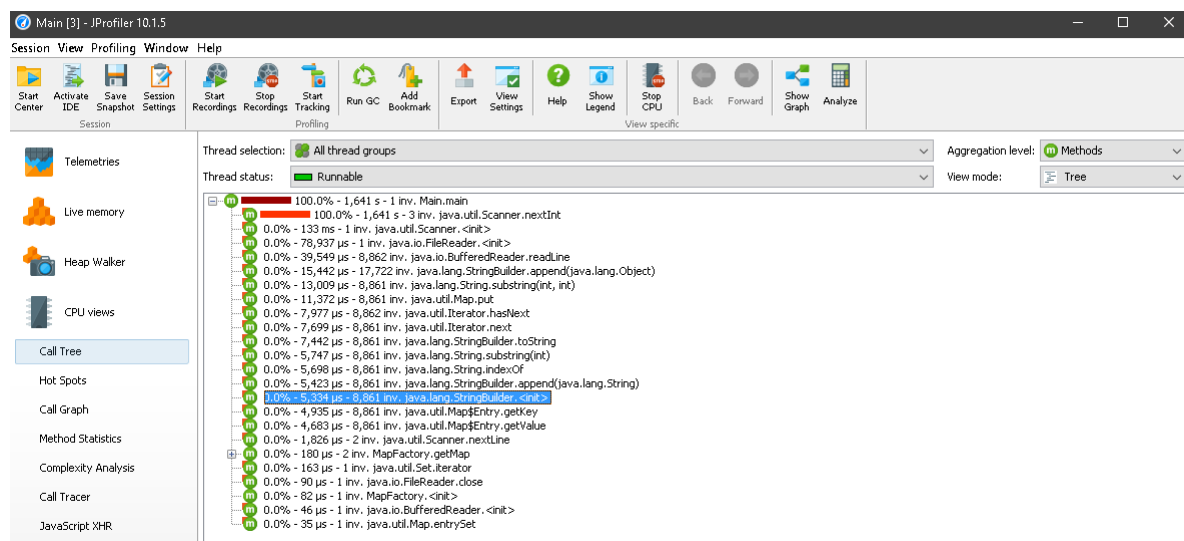


Análisis de Tiempo en Ejecución en JProfiler y Complejidad Calculada

La herramienta utilizada para el análisis de tiempos de ejecución fue JProfiler, la cual se enlazó con IntelliJ y se utilizó el Main para comprobar los tiempos de impresión y ordenamiento de HashMap. Se utilizó la opción 5 del menú (mostrar todas las cartas) para calcular tiempos de ejecución.

Para HashMap

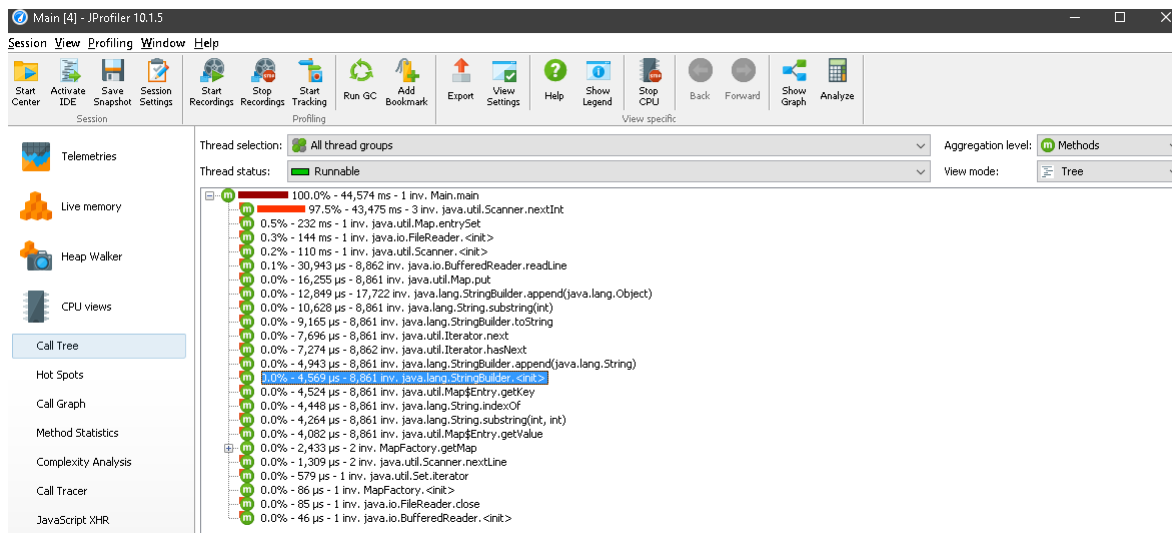
Imagen 1. Tiempos de ejecución para HashMap, StringBuilder y otros métodos.



Tiempo requerido para mostrar: **5334 microsegundos (0.005334 segundos)**

Para TreeMap

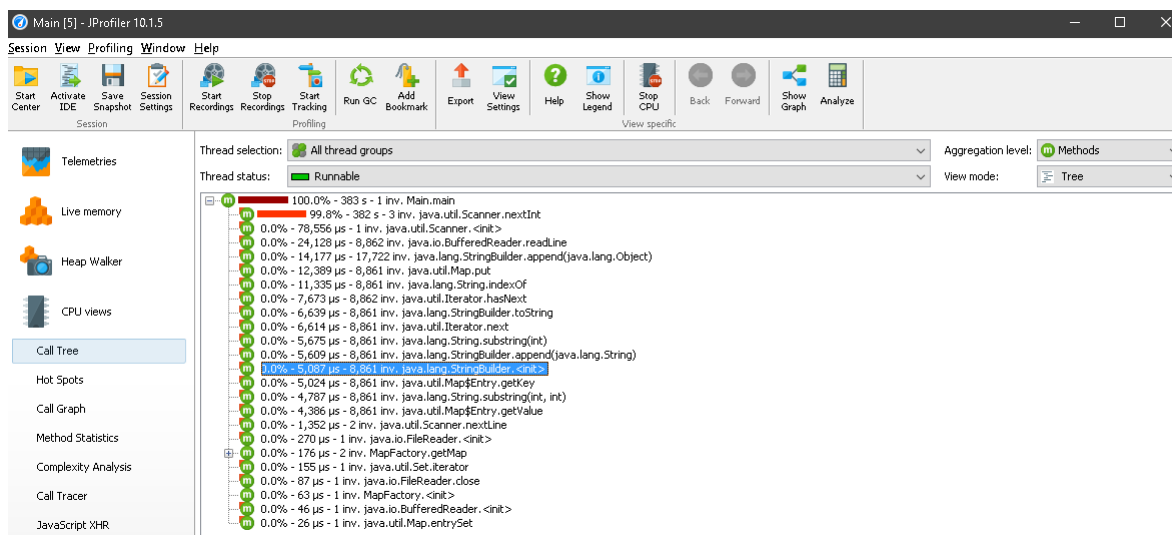
Imagen 2. Tiempos de ejecución para TreeMap, StringBuilder y otros métodos.



Tiempo requerido para mostrar: **4569 microsegundos (0.004569 segundos)**

Para LinkedHashMap

Imagen 3. Tiempos de ejecución para LinkedHashMap, StringBuilder y otros métodos.



Tiempo requerido para mostrar: **5087 microsegundos (0.005087 segundos)**

En conclusión, los tiempos obtenidos por medio de JProfiler indican que la implementación de **TreeMap** a la hora de mostrar los Keys y Values es la más rápida, seguida de **LinkedHashMap** y por último **HashMap**.

Con respecto a la complejidad para la impresión de datos para el caso de HashMap, está representada por $O(n)$, tomando en cuenta el almacenamiento de key/value y métodos de put y get para esta implementación. Siendo n la cantidad de conjuntos de K,V, se ejecutará n veces para los métodos de put/get y así realizar la impresión.