UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Hugo Barillas #23556
Ernesto Ascencio #23009

Hoja de Trabajo 3

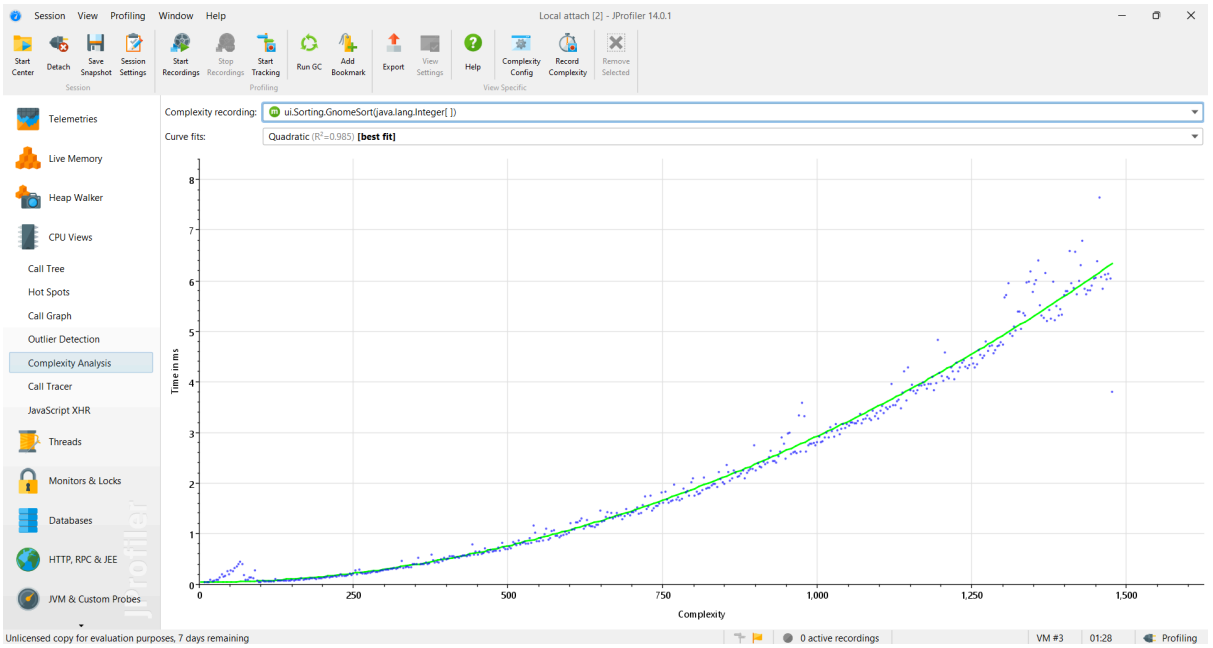
Github: https://github.com/HUG0BA/sorting.git

Profiler Utilizado:

El profiler utilizado es JProfiler que básicamente es una herramienta de java que permite tener un control del tiempo de ejecución de cualquier programa. Este es capaz de dar el rendimiento y su tiempo de ejecución.

Gráficas:

Gnome Sort

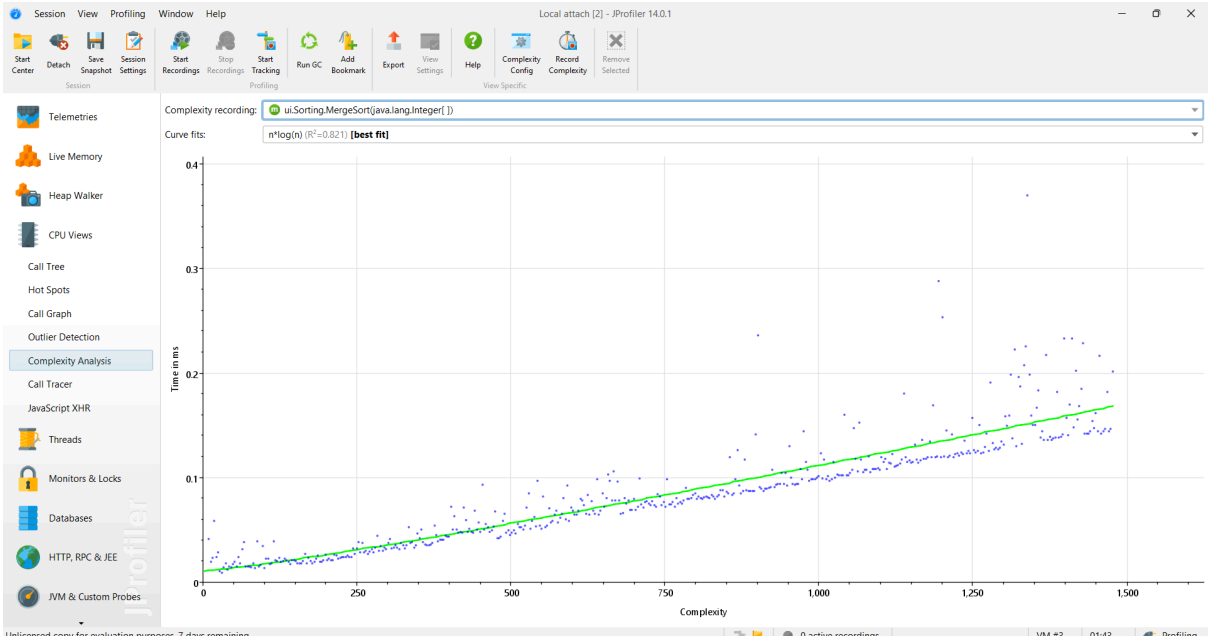


La gráfica demuestra un resultado esperado para el tipo de Sorting utilizado. Debido a que Gnome Sort es un O(n²), si demuestra semejanza con la gráfica de tiempo contra complejidad.

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

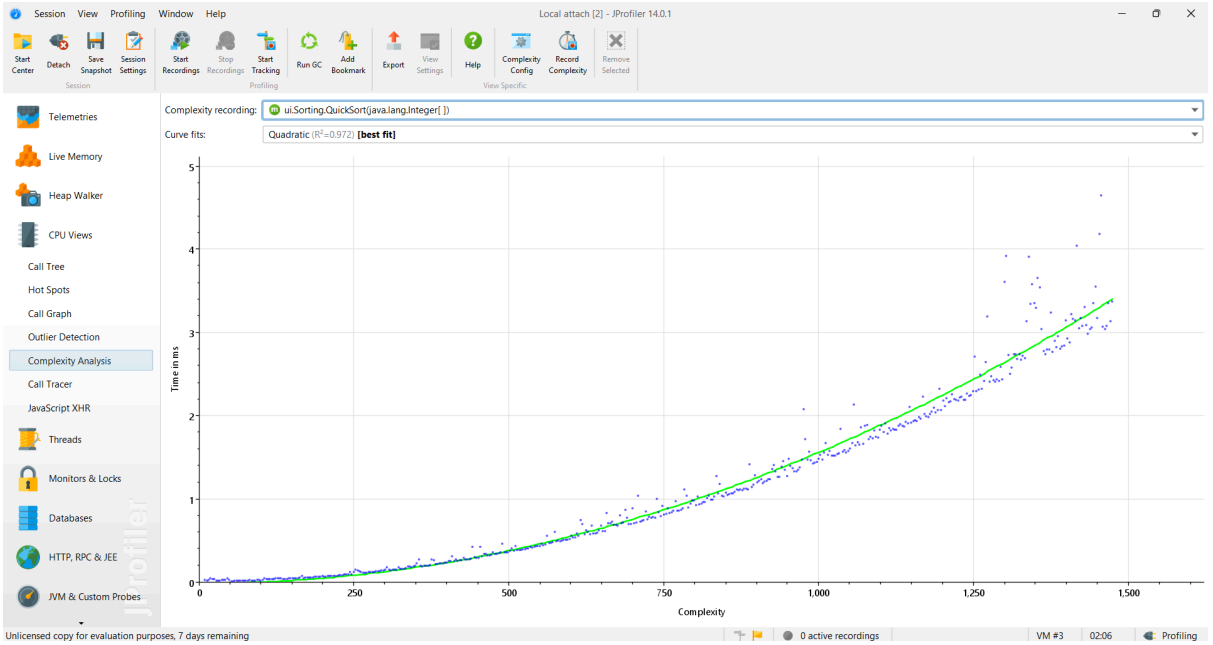
Hugo Barillas #23556
Ernesto Ascencio #23009

Merge Sort



La gráfica de Merge Sort obtuvo un resultado semejante a lo esperado. Esta demostró tener un grado de complejidad según la gráfica de Tiempo contra Complejidad de O(n log(n)).

Quick Sort

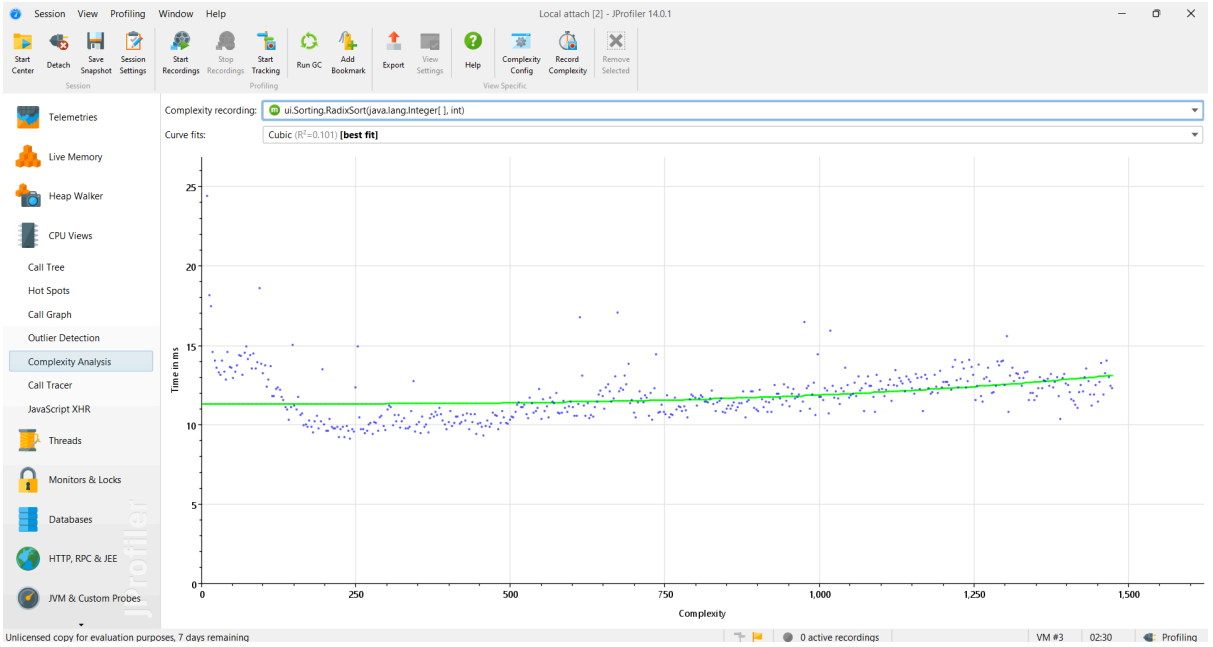


La gráfica del Quick Sort no fue la esperada. Si bien es cierto esta posee dos tipos de escenarios. En el ideal se esperaba un O(n log(n)), sin embargo, según la gráfica de Tiempo contra complejidad, esta tiene un O(n²) posicionándose como un escenario poco deseado.

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

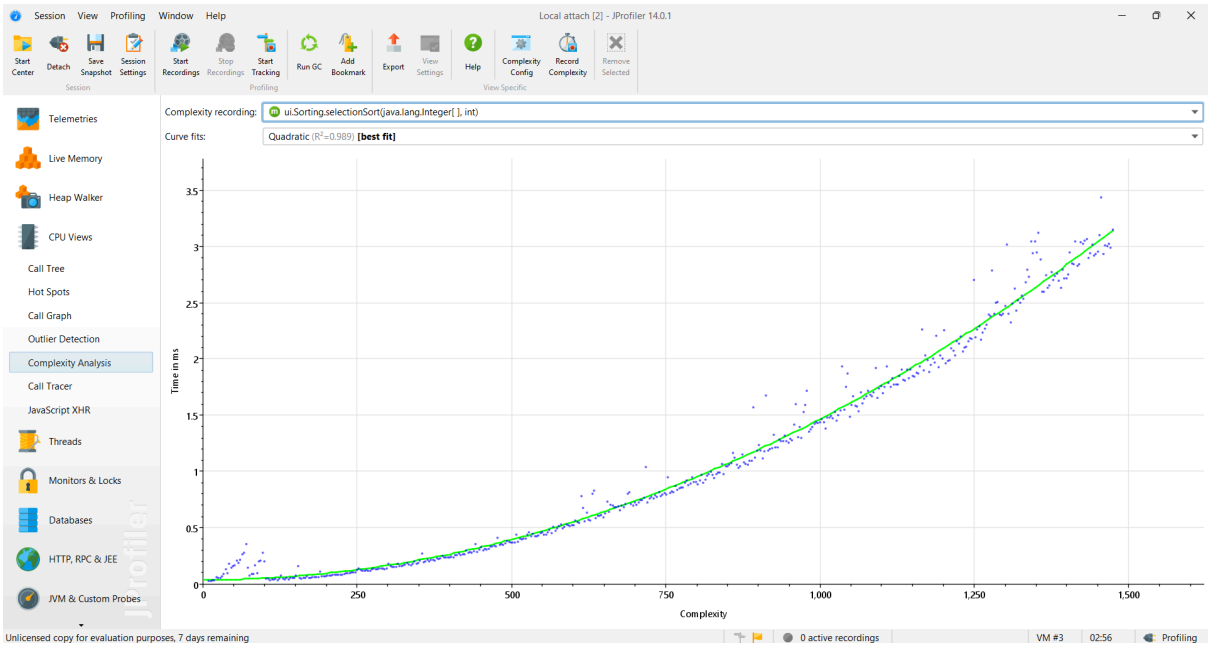
Hugo Barillas #23556
Ernesto Ascencio #23009

Radix Sort



La gráfica presentada por el tipo Radix Sort fue la esperada, a pesar de tener en cierto intervalo una elevación, durante todo su trayecto demostró tener un O(n) el cual fue el esperado.

Selection Sort



El Selection Sort tuvo un resultado esperado según la gráfica Tiempo contra complejidad. Este demostró tener un O(n²) durante todo su tiempo de evaluación.