En el programa se hace uso de la librería time.h y algunas de sus funciones para medir el tiempo de ejecución en la máquina y así tener resultados más acertados sobre la ejecución de los algoritmos.

Heapsort:

- Peor caso: O(n log n)
- Caso promedio: O(n log n)
- Mejor caso: O(n log n)

Heapsort tiene un rendimiento consistente de O(n log n) en todos los casos. Aunque tiene la misma complejidad que Quicksort en el peor de los casos, Heapsort es más predecible y garantiza esta complejidad en todos los casos.

Quicksort:

- Peor caso: O(n^2)
- Caso promedio: O(n log n)
- Mejor caso: O(n log n)

Quicksort es muy eficiente en promedio y funciona en O(n log n) en la mayoría de los casos. Sin embargo, en el peor caso (cuando se elige una mala partición repetidamente), la complejidad puede degradarse a O(n^2). La elección de un buen elemento pivote y la optimización de la partición son fundamentales para obtener un buen rendimiento.

Heapsort es más consistente y predecible, pero generalmente más lento que Quicksort en casos promedio. Quicksort es más rápido en la mayoría de los casos, pero puede ser muy lento en el peor caso.