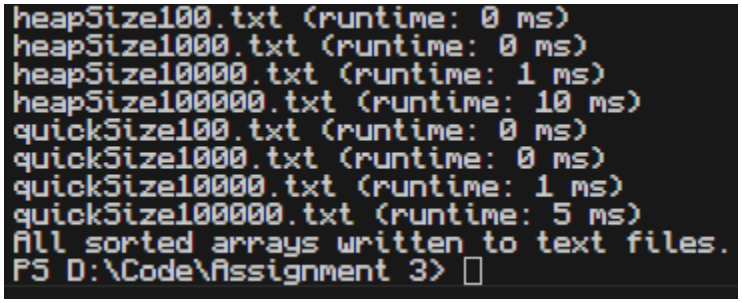
Assignment 3

1. **Compare the performance of heapsort and quicksort based on your experiments.**



Even while Heap Sort also has O(nlogn), it is slower because random memory access has far greater constants. This image shows that Quick Sort is faster because its inner loop has superior cache locality. Remember that although Quick Sort is faster in theory, Heap Sort is still recommended because of the worst-case O(nlogn), but for the average scenario, we would still prefer Quick Sort because of its speed and partitioning effectiveness.

2. **Explain any discrepancies between theoretical and observed runtimes.**

In practice, Quick Sort will perform better and more consistently than Heap Sort because it has smaller constant factors and avoids the worst case of $O(n^2)$; additionally, this partitioning loop optimizes the compiler to run faster and is closest to the average case of O(nlogn). On average, both Heap Sort and Quick Sort have the O(nlogn) case. Heap Sort differs in that it uses a greater constant overhead and does larger calculations without any kind of partitioning loop. As a result, it is typically much slower.