

Connor Dial
CSCI 4270
Dr. Dana
3/21/2025

Assignment 4 Report

1a. Merge Sort:

- Minimum run-time: 9,976
- Maximum run-time: 9,976
- Average run-time: 9,976
- Standard Deviation of run-times: 0

Quick Sort:

- Minimum run-time: 9,568
- Maximum run-time: 499,500
- Average run-time: 10,983.5016
- Standard Deviation of run-times: 647.4450086280995

2. For the minimum runtime of merge sort, which was 9,976 steps on this particular run, it is the best case scenario. It also took slightly more steps on its minimum run-time than quick sort did on its minimum runtime, which was 9,568 steps. This means merge sort performed its sorting slightly slower than quick sort for the minimum runtime.

For the maximum run-time of merge sort, it was also much lower than the maximum run-time of quick sort. Merge sort had a maximum of 9,976 steps, whereas quick sort had a maximum of 499,500 steps. This indicates that quick sort took significantly longer to sort through these arrays than merge sort did.

For average run-time, the difference in steps is smaller with merge sort (9,976) taking much less time to sort the same arrays on average than quick sort (10,983.5016).

3a. It does confirm the every-case time complexity of merge sort. This is because the average-case time complexity for merge sort is $O((n)\log(n))$ where if we plug in 1000 for n , it comes out to around 9976. For our minimum, average, and maximum runtimes, they are also 9976 which serves as confirmation of the every-case time complexity of merge sort.

3b. It does confirm the average time complexity of quick sort. This is because if we use the time complexity of quick sort ($O(n)\log(n)$) and put $n = 1000$ in that formula, this comes out to about 9976, which is extremely close to our measurement 10,983. This confirms it.