ALGO HW3

kwc9ap

October 18, 2021

1 Question 1

For this experiment, we set the input value size, X, to 200. At X = 200, the brute force algorithm took an average (out of five trials) of 2645 microseconds (us). At X = 400, brute force took an average of 10361 us, and 20502 us at x = 600. This is expected, as the time complexity of the brute force algorithm is O(n^2). When the input size X doubles, run-time quadruples. At 3X, the run-time is roughly 9X that of the original input size.

2 Question 2

Again, we set the input value size, X, to 200. At X = 200, the divide and conquer algorithm took an average of 1204 us. When X = 400 and X = 600, the average run-time was 2328 and 3919, respectively. In theory, this D&C algorithm is $\Theta(nlogn)$; the experimental run-times support this. When the input size is doubled from 200 to 400, the run-time doubles (2*log2=2). When the input size is tripled, the run-time theoretically increases by a factor of 4.75 (3*log3). The experimental time quadruples when taken from X = 200 to X = 600. The difference between experimental and theoretical is acceptable could be chalked up to low-level optimization and caching effecting run-times. $\Theta(nlogn)$ is in o(n^2), showing that the D&C algorithm is a big improvement over the quadratic brute force approach.