Jonathan Woolf

861172158

4/21/17

Divide and Conquer vs. Brute Force

Inputs	Seconds (DC)	Seconds (BF)
10	.000128	.0000899
100	.001902	.008399
10000	2.918705	86.99236608
100000	326.1251	11232.657

Based on several tests of each algorithm for each input amount, I concluded that for inputs of 10 or less it was marginally better to use the brute force algorithm. However, my data clearly shows that for large inputs such as 10,000 it was significantly better to run the divide and conquer algorithm. Furthermore, at 100,000 inputs the program takes hours to conclude with brute force. The divide and conquer algorithm succeeded at this amount in mere minutes. Comparing all the inputs will always take longer than dividing the work and using recursion to focus on a more concentrated amount of inputs. Both loops present in brute force have a run time of n. By nesting them we reach a run time complexity of n^2. In the divide and conquer algorithm, the initial sorting has a complexity of n, but the recursive call only has a run time of logn. This leaves the divide and conquer algorithm with a theoretical runtime of $\theta(nlogn)$. With 100,000 inputs, this would be 500,000 or 326 seconds. On the other hand, brute force being a nested for loop has a theoretical run time of $\theta(n^2)$. When n is equal to 100,000, the result is 1E10. This takes over 3 hours proving how massive the disparity in run time complexity is. It is clear both theoretically and in reality, brute force is generally inferior and borderline unusable when compared to divide and conquer.