**Readme:**

I have plotted all 3 graphs for n = 32 to 32768, since 32768 was the max value of n I could get from the previous algorithms. The FFT algorithm could run for much larger values of n (600000+)

I wrote the previous algorithms in C++ using console time to calculate runtime, and I calculated the runtime of the FFT algorithm using the time function in Python, so it got MUCH less average runtime than the previous two. I think if I used the same method to calculate runtime, there would be a cross in the graphs.

**At what values for n does the FFT algorithm cross the two previous algorithms lines?**

The graph does not display a cross point. (Since I had used different ways to calculate runtime in the last 2 problems).

If I use the same method to calculate runtime for all 3 I would have got some crossing point I think, but since they were written in different languages, I couldn’t do that.

**The performance line for the FFT will be nearly linear, why?**

Cause it has a time complexity of O(nlogn).

**Pick a limit on the CPU time (say 10 minutes), what is the maximum problem size each algorithm can solve within the time limit?**

Approximately 600000.