

# Assignment 4 - Fast Fourier transform

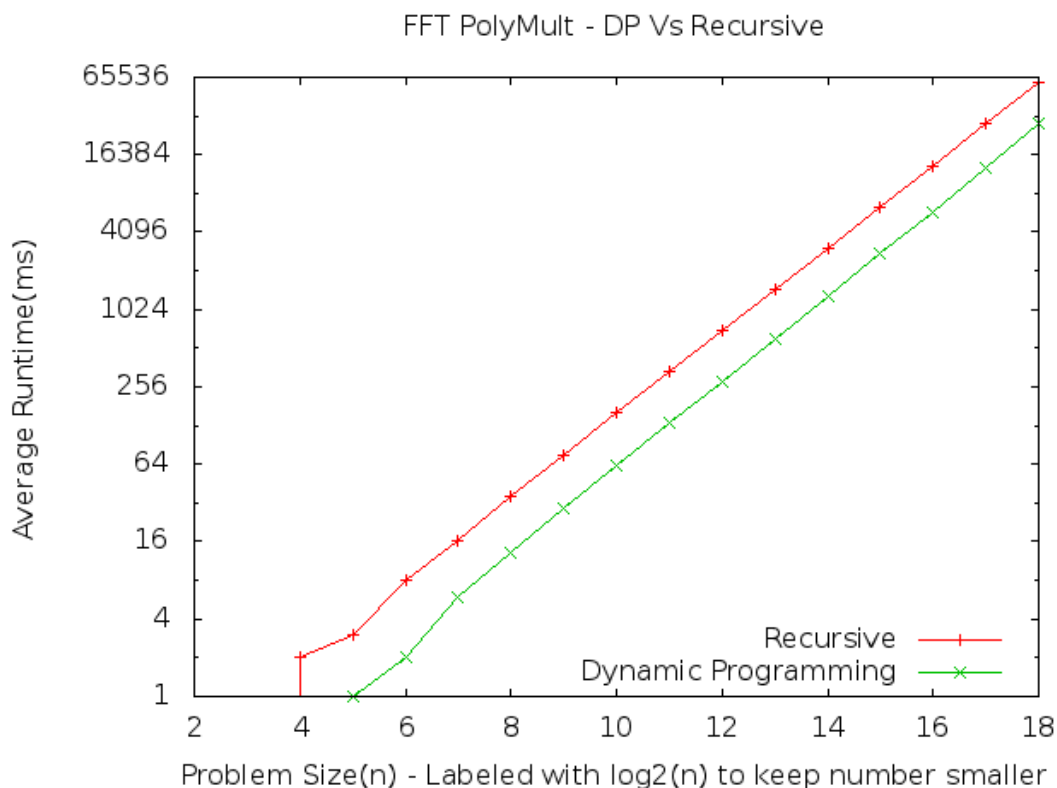
Alan Christensen - 3/24/2014

The following numbers correspond to the numbers in the assignment.

1. I implemented the recursive FFT correctly. Make and run `./FFT test fftworks` to test.
2. To apply the recursive PolyMult FFT method to two polynomials run `./FFT test pmr` and `./FFT test polymultworks`.
3. To check that the computed values are correct and check against older versions of PolyMult run `./FFT test fftvsold`.
4. To apply the recursive DP FFT method to two polynomials run `./FFT test pmd` and `./FFT test polymultworks`.
5. To run timing studies make and run `./FFT run pmr` for recursive and `./FFT run pmd` for DP.

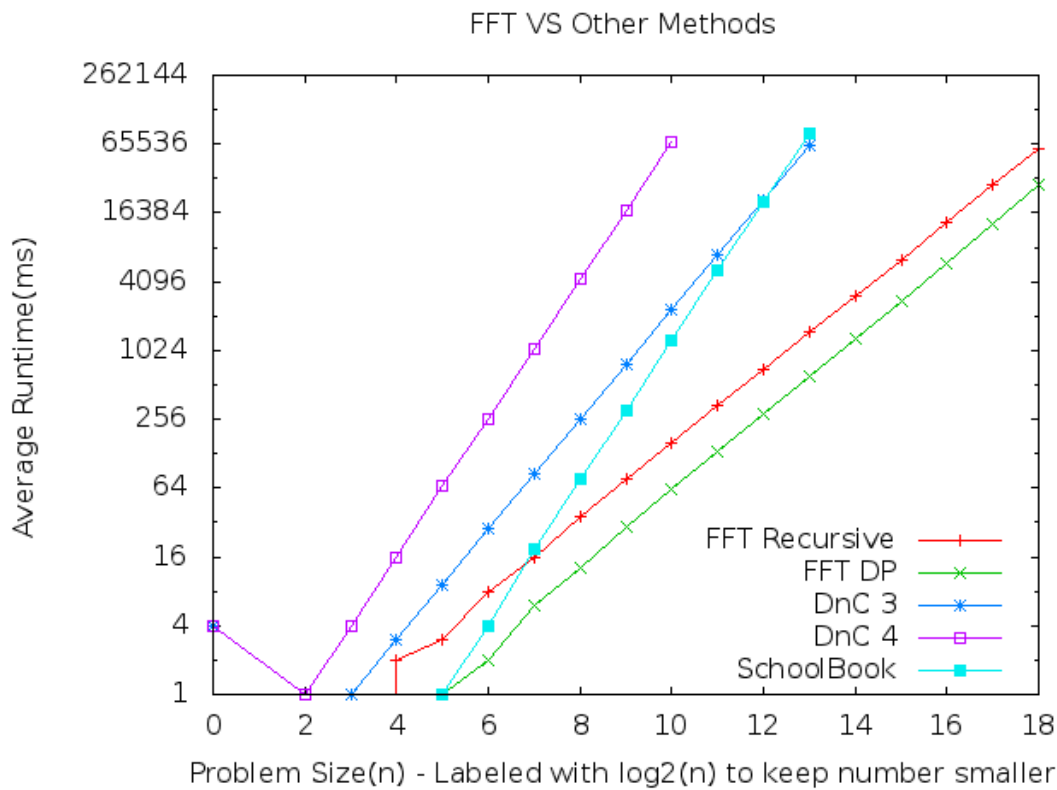
## Graph of the Results

This graph shows the average runtime versus the problem size. The runtime is in milliseconds and the Problem size is shown as  $\log_2(n)$ . The numbers were too big to look good on the graph.



Looking at the plot the recursive seems to take about 2 times as long to compute. It starts out nearer to 3 and comes closer to 2 as the problem size gets bigger.

6) Comparison of FFT vs Other methods we have done.



**Table of the crossing points of the lines**

|       | FFT R      | FFT D      | DnC 3         | DnC 4 | SB |
|-------|------------|------------|---------------|-------|----|
| FFT R | NA         | "          | "             | "     | "  |
| FFT D | ?1         | NA         | "             | "     | "  |
| DnC 3 | $\sim 2^4$ | $\sim 2^2$ | NA            | "     | "  |
| DnC 4 | $\sim 2^2$ | Nope       | Nope          | NA    | "  |
| SB    | $\sim 2^7$ | $\sim 2^6$ | $\sim 2^{12}$ | Nope  | NA |

?1 The recursive seems to be at a slope that is slightly less then the dynamic. So It might cross at very large numbers.

This data shows the the FFT is in fact faster. I think there are possible optimizations that would make some run faster than others at times. But in the end the slope is the main concern in this analysis.