**Assignment 4**

Gattu Varun A02092613

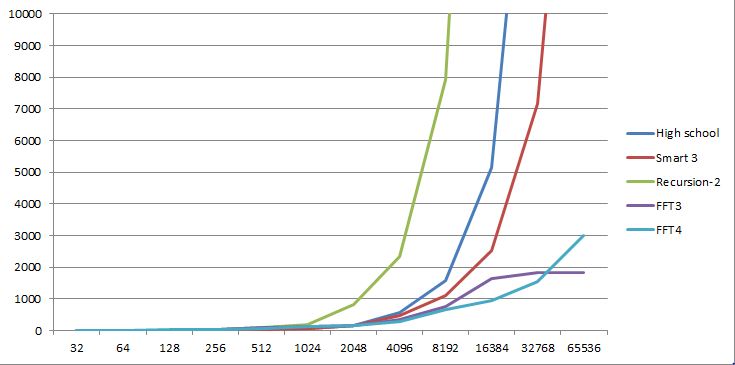
Fast Fourier transformation algorithm is used here to multiply two complex numbers. This algorithm uses the complex number instead of Integers or Doubles. Complex numbers have a real part and an imaginary part. The results of using FFT with 3 real multiples and 4 real multiples are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | High school | Smart 3 | Recursion-2 | FFT3 | FFT4 |
| 32 | 3 | 2 | 4 | 9 | 8 |
| 64 | 7 | 5 | 8 | 17 | 12 |
| 128 | 16 | 18 | 13 | 17 | 24 |
| 256 | 30 | 23 | 32 | 30 | 30 |
| 512 | 58 | 47 | 95 | 95 | 82 |
| 1024 | 74 | 69 | 193 | 123 | 117 |
| 2048 | 159 | 149 | 823 | 160 | 169 |
| 4096 | 563 | 477 | 2340 | 364 | 300 |
| 8192 | 1592 | 1113 | 7917 | 775 | 664 |
| 16384 | 5152 | 2517 | 32552 | 1649 | 960 |
| 32768 | 20101 | 7165 | 119819 | 1840 | 1548 |
| 65536 | 128146 | 23348 | 484467 | 1829 | 2996 |

This algorithm has a very good performance for input values greater than 2048. For smaller inputs, it gives a performance lesser than or equal to Smart three algorithm but after 2048 input size, FFT gives the best possible results.

We implemented FFT with 3 real multiples and 4 real multiples – FFT3 and FFT4 respectively. FFT4 gives the best results in fewer time compared to FFT3 for any input size.

The following graph gives a better insight to the performance of the algorithms:



I have rounded the Y-Axis (time taken) to 10,000 for a clear picture and a better understanding.