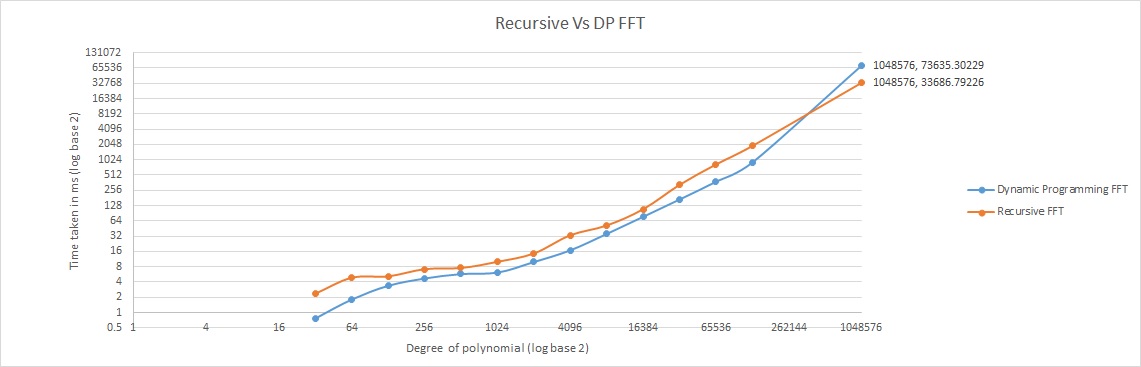
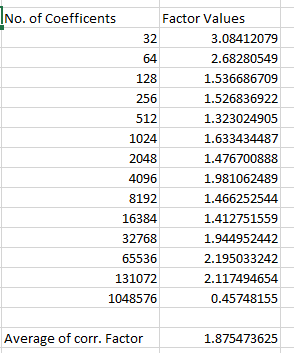
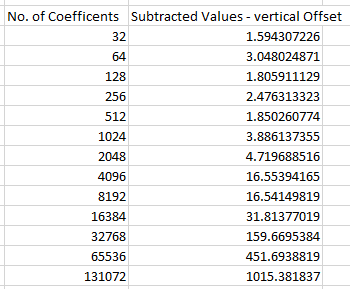
**Technical Explanation of Algorithms:**

**Question – 5:**

****

Graph between Dynamic Programming FFT and Recursion. Dynamic programming is generally as expected perform better that recursive. Since there are no recursive calls and no overhead time. But one surprising thing about this experiment is for no of coefficients = 1048576 the recursive preforms better than the dynamic.



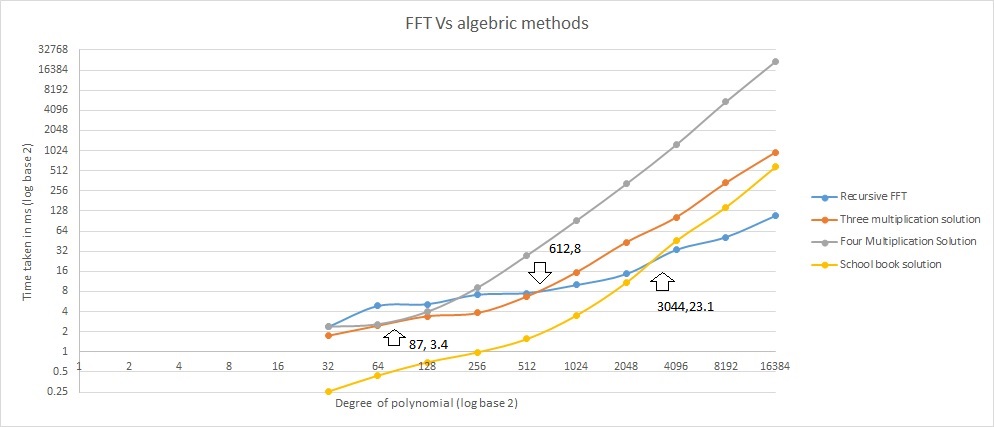
The first table gives the values of subtracted vertical offset – real values between the recursive and dynamic programming, as you can see from the graph this increase with the increase of the no of coefficients. So dynamic programming performance increase with increase of number of coefficients.

Second table gives us the corresponding factor and this also increase with the increase of number of coefficients and the average we get is 1.875 times the recursive algorithm.

How much faster is DP compared to recursive? -

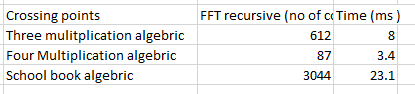
Ans – 1.875 times (in average)

**Question 6:**

****

This gives the performance between the Recursive FFT and other three algebraic methods that are done in the last assignment. And the cross over points are mentioned in the graph as shown above

First cross over point happened by four multiplication followed by three multiplication, school book. The cross over points table is as follows



As can be observed, even though other algebraic methods have good performance for smaller number of coefficients, the recursive FFT out performs the algebraic methods for larger coefficient values.

**Question 7:**

The above graph is between the mean absolute error between the FFT solution and three multiplication solution, we can see that as the number of coefficients increase the mean absolute error is also increased. This quite expected as the number of coefficients increase the number of terms increase in the final multiplication resulting in more mean absolute error.

**Question 9:**

The performance of the complex multiply using 3 real multiplies and one where it using 4 multiplies is of a little surprise as the 3 multiplies is expected to be faster but in fact I got it reverse.

The average time of using 4 multiplies is - 73635.30229 ms

The average time of using 3 multiplies is - 77215.53422 ms

So, this is a bit of surprise :D