# Assignment 7

Inderpreet Singh Chera - 160101035

## 2 Polynomial Multiplication:

This multiplication of polynomials is carried out by 3 methods:

1. School Method
2. Karatsuba's Algorithm
3. FFT divide and conquer

### System Settings

All tests were done on **Intel(R) Core(TM) i3-5005U CPU @ 2.00GHz** processor. This computer is **dual core** where each core has **2 threads**. Also during each experiment it was insured that **no other applications** were running in the background, so that we don't have any biased readings.

### Implementations

We have 2 polynomials, say a and b with n-1 max power, i.e., the total number of elements in each polynomial is n.

#### School Method

This is a naive way of multiplying 2 polynomials. We multiply each element of b with all elements of a and get the output. This takes O(n) time complexity.

To do the same in parallel and to distribute almost the same amount of work to each processor, we tell each processor to get the answer of each mod p, which has value equal to processor id, coefficient of final aray. If p divides n, then it can be shown that the same amount of work is done by all the threads.

#### Karatsuba's Algorithm

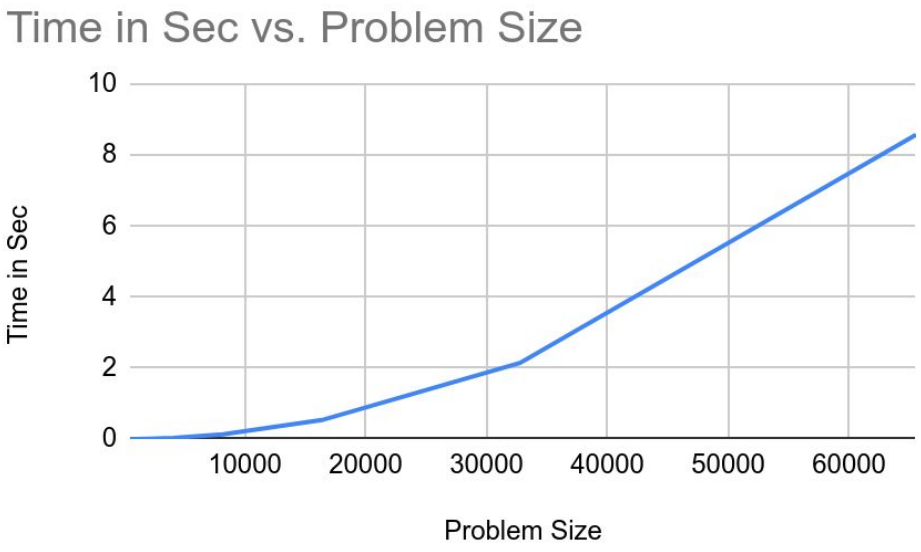
#### FFT Divide and Conquer

# Observations

## School Method:

1. Changing problem size and using 4 threads:

| Problem Size | Time in Sec |
|---|---|
| 512 | 0.000733229 |
| 1024 | 0.00241978 |
| 2048 | 0.00911779 |
| 4096 | 0.0333559 |
| 8192 | 0.13334 |
| 16384 | 0.535747 |
| 32768 | 2.14428 |
| 65536 | 8.57661 |




2. Changing Number of threads keeping size = 65536:

| Num Threads | TIme (in sec) |
|---|---|
| 1 | 18.8415 |
| 2 | 9.44706 |
| 4 | 8.56408 |
| 8 | 8.54682 |
| 16 | 8.5623 |