

Documentation Lab 5 (Parallelizing techniques) - Mos Daniele 935

Algorithms:

Regular algorithm $O(n^2)$

How does it work:

- We distribute each term of the first polynomial to every term of the second one
- Multiply the coefficients and then add the exponents
- Get the sum of the terms resulted from the previous multiplications which result in the same exponent

Karatsuba $O(n \log n)$

How does it work:

- It is a more efficient way of multiplication
- Uses divide and conquer to divide the given numbers in two halves. Let the given numbers be X and Y.

```
X = Xl*2n/2 + Xr    [Xl and Xr contain leftmost and rightmost n/2 bits of X]
Y = Yl*2n/2 + Yr    [Yl and Yr contain leftmost and rightmost n/2 bits of Y]
```

Tests:

- tests are done using 5 threads
- the time is measured in milliseconds

Degree	Simple Sequential	Simple Threaded	Karatsuba Sequential	Karatsuba Threaded
5	3	19	2	14
10	2	10	2	28
20	2	9	4	59
50	4	8	12	72
100	7	12	22	75
500	39	43	58	141
1000	73	78	295	216

Conclusions after testing:

- we can see that Karatsuba sequential did better in smaller cases, but way worse in cases where the degree of the polynomial is larger
- if we use simple threaded we can observe that the execution time increases so it is not so good