📄

Mos Daniele 935 MPI programming Lab 7 Documentation

Algorithms:

Regular algorithm *O(n2)*

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(nlogn)*

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]
```

Distributed algorithm using MPI:

We execute the following steps in order to distribute the work between nodes:

- We first divide the polynomial’s length by the worker’s number
- Need to keep in mind the exclusion o the master process
- We compute the operations accordingly
- We use the ‘send’ function from MPI in order to send data to a certain node
- The node computes the operation
- Then the node sends back the result
- The master process has to add up all the partial results from the worker nodes in the end in order to get the final result of the multiplication

Tests from current MPI runs:

- the time is measured in milliseconds

Degree	Simple using MPI	Karatsuba using MPI
5	88	76
10	82	63
20	68	68
50	92	83
100	84	107
500	180	219
1000	264	348

Tests from previous tests from regular CPU (assignment 5):

- 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