Parallelizing Techniques

# Goals

The goal of this lab is to implement a simple but non-trivial parallel algorithm.

# Requirement

Perform the multiplication of 2 polynomials. Use both the regular O(n^2) algorithm and the Karatsuba algorithm, and each in both the sequential form and a parallelized form. Compare the 4 variants.

# Computer Specification

\* CPU: Intel Core i7  
\* RAM: 16 GB  
\* System type: 64-bit

# Short Description of the Implementation

Algorithms:  
 \* Regular polynomial multiplication  
 \* Karatsuba algorithm

## Regular Polynomial Multiplication

Complexity: O(n^2)

Step 1: Distribute each term of the to every term of the second polynomial. Remember that when you multiply two terms together you must multiply the coefficient (numbers) and add the exponents.

Step 2: Combine like terms (if you can).

## Karatsuba Algorithm

Complexity: O(n^log3)

A fast multiplication algorithm that uses a divide and conquer approach to multiply two numbers.

# Performed Tests

Note: By level "x" I am referring that the algorithms were used to multiply 2 polynomials of rank x and x - 2, with coefficients being random numbers between -10 and 10

# Tables

|  |  |  |  |
| --- | --- | --- | --- |
|  | Level 8 | Level 50 | Level 100 |
| Regular Sequential | 11 ms | 21 ms | 32 ms |
| Regular Parallelized | 5 ms | 9 ms | 12 ms |
| Karatsuba Sequential | 0 ms | 0 ms | 3 ms |
| Karatsuba Parallelized | 0 ms | 0 ms | 0 ms |

# Conclusion

For the most part, the parallelized versions of the algorithms run faster.

Karatsuba’s algorithm is clearly superior to the regular algorithm, and for large numbers, it would be preferred.