- ***The algorithms:***
  - **Regular polynomial multiplication:**

    Take each term one by one of the first polynomial and multiply with every term of the second polynomial. Also add them when the terms have the same power. It has a time complexity of O(n^2).
  - **Karatsuba multiplication algorithm for polynomials:**

    Uses Divide and Conquer to divide the polynomial into smaller polynomials that can be easily solved so that the whole multiplication process is heavily reduced. We break the polynomial until we reach polynomials of order 0 or 1 and those we just multiply since it requires, at most, 4 multiplications on generally smaller numbers.

- ***The synchronization:***

For the regular polynomial multiplication, we use a ThreadPool with a fixed number of threads. Each task run has as parameters the 2 polynomials and an interval of indexes for which that specific task will calculate the result for.

For the Karatsuba algorithm, we also use a ThreadPool with a fixed number of threads. Then the usual recursive calls of the algorithm will become threads having as parameters the smaller polynomials that need to be multiplied. This will be repeated until the polynomials are small enough to be calculated (order 0 or 1).

- ***Results:***

	size 100/ 5 threads	size 1000/ 10 threads	size 5000/ 25 threads	size 10000/ 50 threads
Regular	0.00099	0.09976	2.43901	9.64327
Regular parallelized	0.00402	0.21079	4.51095	16.83761
Karatsuba	0.00499	0.10778	2.31173	6.15926
Karatsuba parallelized	0.00398	0.17631	5.18134	19.05509