

Complexity comparison of Karatsuba, divide and conquer and grade school algorithms

Author: Oleg Ivanov 193.1

Supervisor: Alexey Varenikov

Submission date: 26th April 2020

Problem statement

Algorithm's difficulty:

- Grade school multiplication: $O(n^2)$
- Divide and conquer multiplication: $O(n^2)$
- Karatsuba multiplication: $O(n^{1.6})$

Algorithm's principle:

- Grade school multiplication: $x \times y$
- Divide and conquer multiplication: $x \times y = (a \times 10^{n/2} + b) \times (c \times 10^{n/2} + d) =$
 $= ac \times 10^n + ad \times 10^{n/2} + bc \times 10^{n/2} + cd$
- Divide and conquer multiplication: $x \times y = (a \times 10^{n/2} + b) \times (c \times 10^{n/2} + d) =$
 $= ac \times 10^n + ad \times 10^{n/2} + bc \times 10^{n/2} + cd = ac \times 10^n + (ad + bc) \times 10^{n/2} + cd = ac$
 $\times 10^n + cd + ((a+b)(c+d) - ac - bd) \times 10^{n/2}$

I've decided to use `std::string` as a holder of our numbers. In my class `Number` the only field is string variable. I have a lot of overloadings, such as `+`, `-`, `*` and so on to make my project better to read and work with.

Useful data:

1. [Karatsuba's algorithm](#)
2. [Divide and conquer algorithm](#)
3. [Grade school multiplication](#)

Code

Structure:

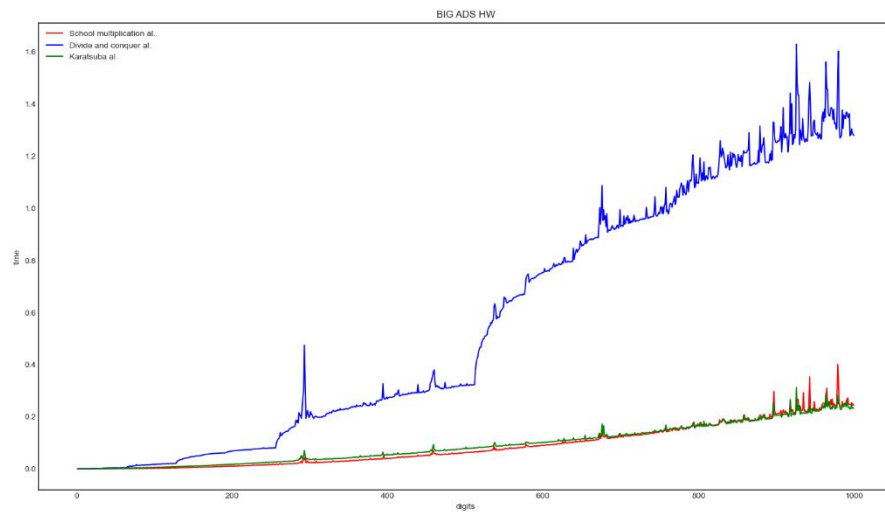
1. main.cpp
2. Multiplicator.cpp
3. Number.cpp
4. Multiplicator.h
5. Number.h

In header files I have only function headers and classes. The implementation of functions is situated in the .cpp files with the same name. In my recursive algorithms implementations the base cases are all 1. I've tried to avoid any sort of multiplication, that is why I made the static second field (`std::vector<std::vector<std::string>>`). It is a multiplication table for 1-digit integers. Also, I use smart pointers working with polymorphic classes. In main.cpp I have two functions. Their main goals are to execute the algorithm 3 times and then write in csv file. Then by using python I build 3 graphs to represent asymptotics. Pandas and matplotlib helped me with the realization of graph.

URL:

<https://github.com/olivan139/dsba-ads2020-hw1>

Results



- School multiplication al.
- Divide and conquer al.
- Karatsuba al.

As it can be seen from the graph divide and conquer algo is slower than grade school algo because of const factor. Recursion makes it slower.

Conclusion

To sum up, despite the fact that I faced a lot of problems during implementation I still like this task. The only thing I would improve is divide and conquer algorithm implementation.