Complexity of different multiplication algorithms.

Author: Fishman Max 191.1

Supervisor: George P.

Submission Date: 26.04.2020

In this section I will tell you what I encountered during the writing of the project.
And so, since we need to implement multiplication algorithms, I first found out what difficulties the algorithms have.

Grade school multiplication: O(n^2)

Karatsuba: O(n^log(2 3)); T(n) = 3T(n/2) + O(n) where O(n) includes + and - => T(n) = O(n^1.6)

Divide and Conquer: O(n^2)

Formulas which I used in my project for algorithms:

School Grade Multiplication: I multiplied the digits of each number by multiplying the sum by 10 to the power of the indices and added the resulting numbers, this is a simple school algorithm for multiplying.

Divide and Conquer: xy = (a*10^(n/2) + b) (c*10^(n/2) + d) = ac*10^n + (ad + bc)*10^(n/2) + bd.

Where: x = a*10^(n/2) + b; a, b – halves of x

y = c*10^(n/2) + d; c, d – halves of y

Karatsuba:

It is clearly that Kratsuba is derived from the divide and conquer algorithm, so that he has reduced the number of multiplications that reduce the execution time of the algorithm

So xy = ac*10^n + (ad + bc)*10^(n/2) + bd; (ad + bc) = (a+b)(c+d) - ac – bd, where we already calculated ac and bd.

So its complexity is O(n^16)

In conclusion. First of all we need to decide what we will store numbers in and how, it's more convenient for me to store numbers in the string type and store them in reverse order, later we will return them to normal. Then We will overload the addition and subtraction operators to correctly subtract the numbers and implement the multiplication algorithms, after which we create the test_up_to function that runs our algorithms and displays the time of their execution. Next, we will deal with the virtual environment and how to build graphs in python. It is General Plan.

List of useful literature:

1) https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm

2) https://en.wikipedia.org/wiki/Karatsuba_algorithm

3) https://en.wikipedia.org/wiki/Multiplication_algorithm

4) https://wiki.archlinux.org/index.php/Python/Virtual_environment

5) http://www.cplusplus.com/reference/ctime/

CODE.

Header.h

1) first thing i create class Number in private, I store numbers in type string in reverse order.

In public I have such functions:

```
1.1) remove_zeroes() which allow me to delete all '0' in the beginning of string
1.2) to_str()
1.3) print()
1.4) make_number_raw()
For example: from string "123" it makes number 123
1.5) make_number()
For example: from string "123" it makes number 321
1.4 and 1.5 are done for easier multiplication actions.
1.6) split() which splits our number by 2 for Karatsuba and Divide and Conquer
1.7) shift() // *= 10^power
1.8) size()
```

And such things as operators overload for: ==, !=, +,-, +=, -=.

2) Class Multiplicator in private which I have check for the correctness of the multiplication, i.e. the transition to exceptions. And in public:

```
2.1) school_grade()
2.2) divide_and_conquer()
2.3) Karatsuba()
2.4) what() it's a method for each exception
```

in all three algorithms with a number in which <= 10 digits I call school grade to reduce time, George allowed us to use this to optimize algorithms.

```
2.5)it's a function of creation a random number
 static Number random(size_t size)
{
   std::random_device rd;  // используем энтропию системы для получения сида
   std::mt19937 gen(rd()); // создаем генератор с полученным сидом
   std::uniform_int_distribution<> dis(0, 9); // задаем диапазон, в котором будем получать числа (цифры
0-9)
   std::string digits;
   for (size_t i = 0; i < size; ++i)
```

```
    digits.push_back(dis(gen));

  return {digits};
}

    digits.push_back(dis(gen));

  return {digits};
}
```

```
2.6) test_up_to() to test our algoritms and find the time for that they are working for in seconds using <ctime>
```

Main

In main I just I make a conclusion to the csv file in the format of size, school grade, D&C, Karatsuba and run these algorithms for finite numbers of size n from the given initial numbers and the given step between the numbers, so that the result can be seen faster.

Python

At its core, the main goal of the Python virtual environment is to create an isolated environment for Python projects. So by including venv in my project I create a virtual environment and construct Graphs by using

```
import pandas as pd
import matplotlib.pyplot as plt
```

Finally url:

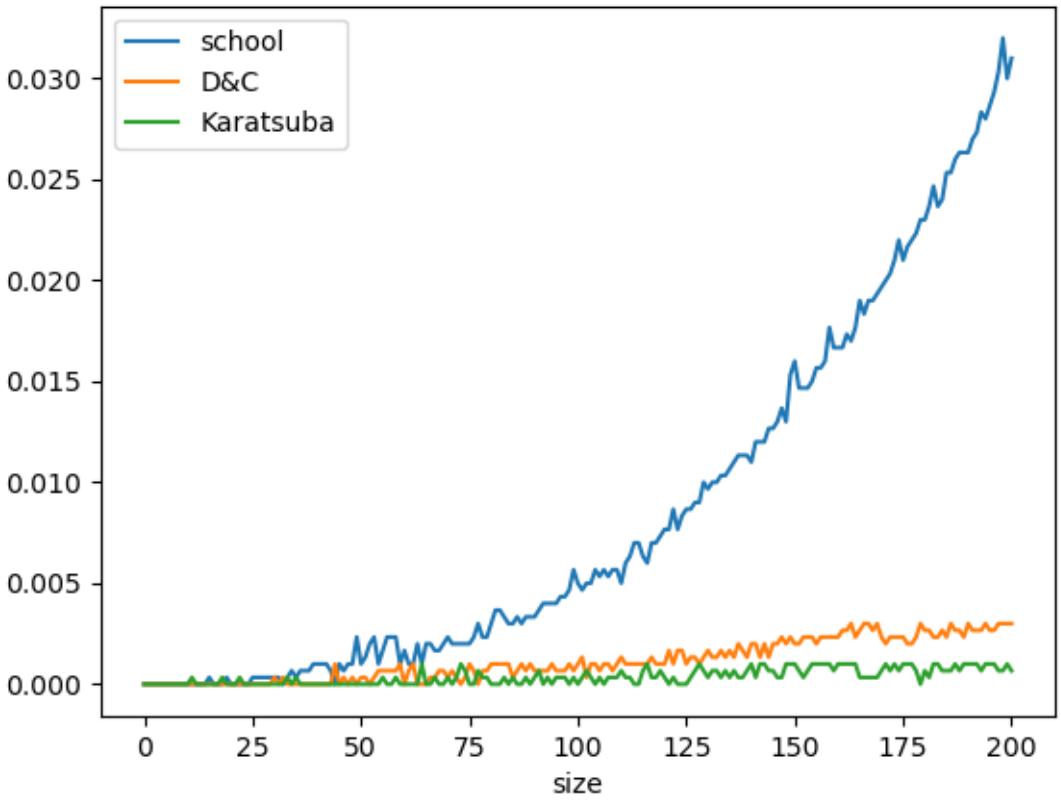
Results

1)for n=200 first=0 step=1;

```
Grade school: 0.031
```

```
Karatsuba: .000666667
```

```
D&C: 0.003
```

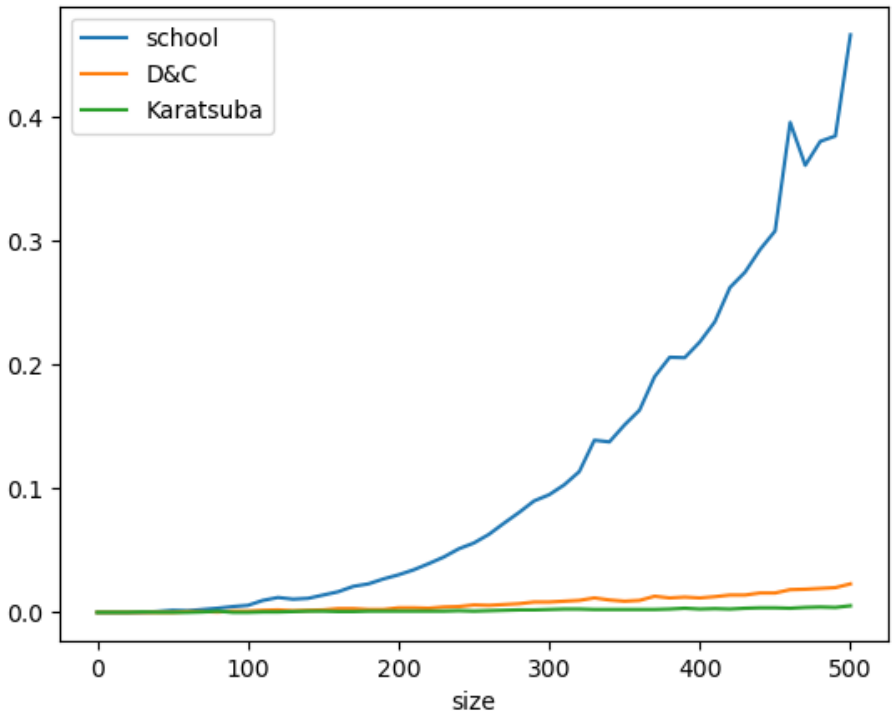


2)For n=500 first=0 step=10;

```
School Grade: 0.466333
```

```
Karatsuba: 0.00533333
```

```
D&C: 0.023
```

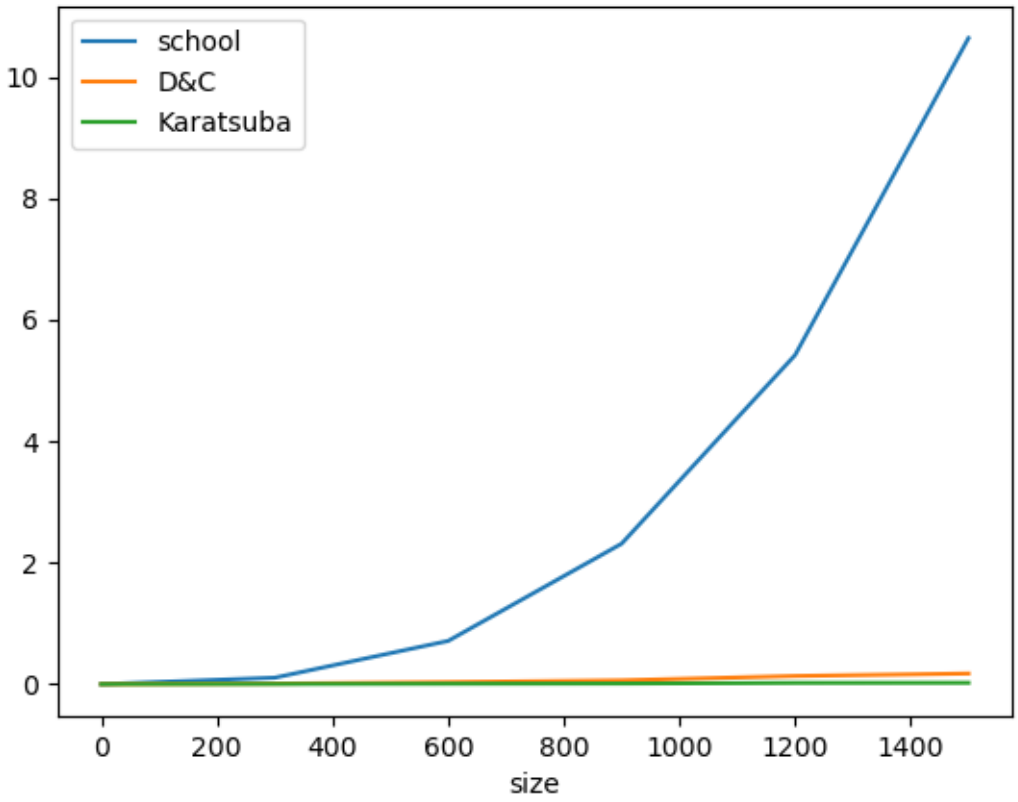


3)n=1500 first=0 step=300

```
Grade School: 10.6367
```

```
Karatsuba: 0.0226667
```

```
D&C: 0.175
```



The D & C algorithm works faster than in elementary school, but all in the same sense O(n ^ 2) due to the implementation of its algorithm in the code. In general, I am satisfied with the results, they correspond to what I wrote at the beginning, and the generation of random numbers works fine, so that there will be no identical results with the same parameters.

Conclusion

research work carried out. According to my project, you can watch the time for which 3 algorithms and their schedules are executed. I would improve the external structure of the project, namely, make more .h files, which would improve code-style and come up with a different implementation for the D&C algorithm.