HSE DSBA

BIG HW # 1

Multiplication algorithms implementation and working time measuring

Emil Fakhretdinov

26.04.2020

Supervisor name Rudakov Kirill

Group number 192-1

Problem statement section

Plan

1. Implement Grade School Multiplication Algorithm, Divide and Conquer Multiplication, Karatsuba Multiplication
2. Measure working time on different number sizes
3. Compare results
4. Implement python graphs using obtained results
5. Upload on GitHub

The problem I've learned that different algorithms that do the same thing, just multiply two numbers can have different working time due to their complexity.

First of all I have to claim that all algorithms are based on Grade School Multiplication.

Theoretically Grade School Multiplication complexity is T(n) = $\theta(n^2)$

This is due to the fact that algorithm consider two numbers of size n and multiply each digit from first number with each digit from second number

The next algorithm is Divide and Conquer theoretically it takes T(n) = $\theta(n^{\log 3})$

This is due to the fact that this algorithm divide big problem into smaller subproblems, it is always easier to compute small numbers than long. It has 4 recursive calls.

Karatsuba approach is an upgraded DaC algorithm it's runtime T(n) = $\theta(n^{\log 3})$, but on long numbers it works faster because we recursively call it 3 times whereas DaC 4 times.

literature sources:

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

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

Lectures

Implementation details section

DaC

```
Number ac = DaC(pair_num1.first,pair_num2.first);
Number ad = DaC(pair_num1.first,pair_num2.second);
Number bc = DaC(pair_num1.second,pair_num2.first);
Number bd = DaC(pair_num1.second,pair_num2.second);
```

Karatsuba

```
Number a = Karatsuba(pair_num1.first,pair_num2.first);
Number d = Karatsuba(pair_num1.second,pair_num2.second);
Number e = Karatsuba(pair_num1.first + pair_num1.second, pair_num1.first +
pair_num1.second) - a - d;
```

This is why Karatsuba is faster

To measure time I've used

```
#include <ctime>
clock_t start = clock();  To start timer
clock_t end = clock(); To end timer
```

By the way, in DaC and Karatsuba it is better to change base case to constant 64 because with GSM and constant 64 it will work faster.

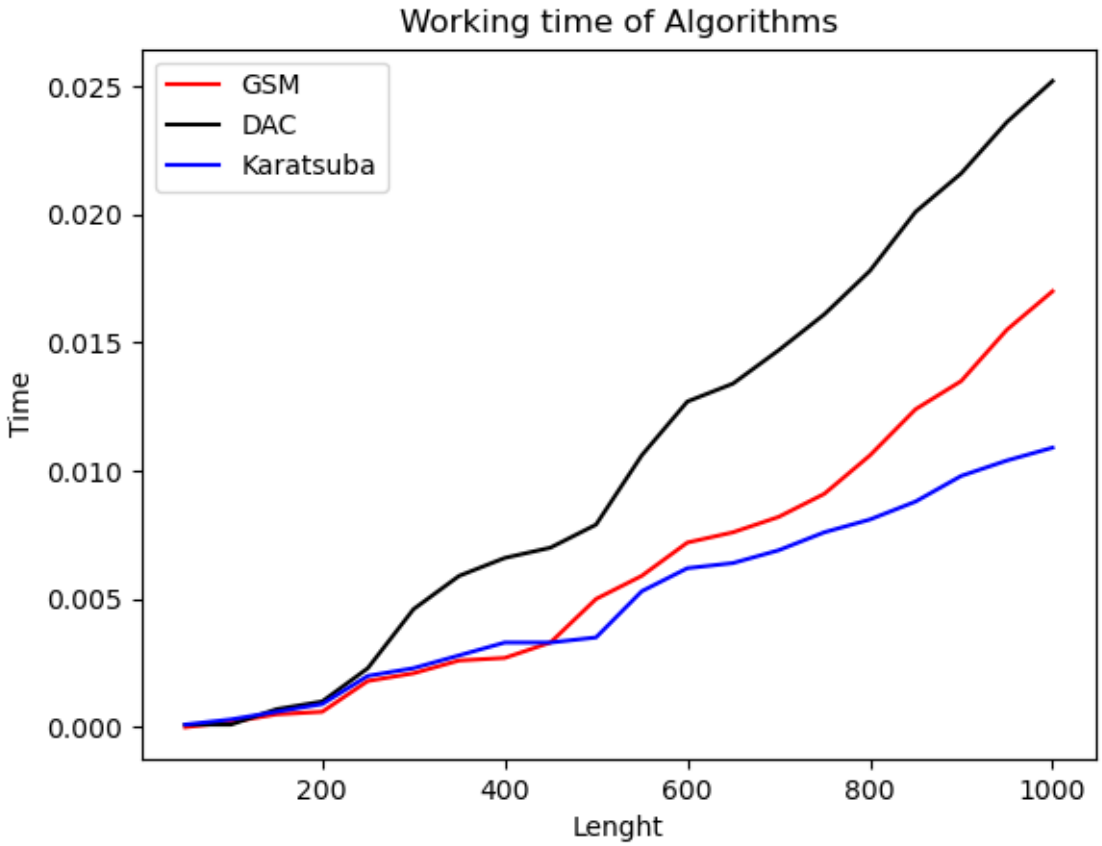
No any other specific approaches were used. Only standard c++ tools.

GitHub URL: https://github.com/eliray01/dsba-ads2020-hw1

Results and discussion

After dealing with code I've obtained results.

Here is the graph of run time of algorithms



As we can see before size 300 they have nearly same runtime, but then everything changes.

DaC is working slower due to the fact that it has a lot of dynamic memory allocation and in my case I lot of reverses of numbers.

So the results are DaC is the slowest it takes 0.025 sec to deal with 1000 digit numbers.

The next is GSM and it takes 0.016 sec to deal with 1000 digits.

And the fastest way to multiply 2 numbers of size 1000 is Karatsuba multiplication it takes only 0.01 sec to deal with this problem.

My obtained graphs and results have some small deviation from theoretical results, due to different pc(processor), but they are almost the same.

Conclusion

To conclude, I would say that obtained results agreed with theoretical results, with small deviations. For further improvement of the work I would think how to improve DaC algorithm.