**ALGORITHM ANALYSIS**

By: Steve George Parakal

Group: 193-1

Submission Date: 26th April 2020

Supervisor: Aleksey Varenikov

Problem Statement

To implement a program to estimate and compare the graphs of three multiplication algorithms, based on datasets generated by calculating time taken for algorithm to get product of randomly generated numbers of different digits.

The algorithms are namely: Grade School Multiplication Algorithm, Karatsuba Algorithm and Divide and Conquer Algorithm.

The theoretical complexities for the algorithms are the following:

Grade School Multiplication Algorithm it is

Karatsuba Algorithm it is

Divide and Conquer Algorithm it is.

Formulas Used:

**Divide and Conquer Algorithm**:

Where and a, b are two halves of x

and c, d are two halves of y

**Karatsuba Algorithm**:

But

Where and a, b are two halves of x

and c, d are two halves of y

Citations

* <https://en.wikipedia.org/wiki/Multiplication_algorithm>
* <https://en.wikipedia.org/wiki/Karatsuba_algorithm>
* <https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm>

Implementation Details

Within the program a class named Number has been implemented in a header file Number.h which acts as a custom data type for storing large numbers, it has a private member field which is a string. Its public member functions include one that returns a character at an index, one that returns size of string, it also includes a function that splits a string and returns a pair containing both parts, a function that appends zeroes to the string and returns it. There are also two overloadings for addition and subtraction operators as well as a return function for the private member string.



Another class Multiplicator in header file Multiplicator.h (which calls the Number.h header file) has been implemented as well as an abstract class with two public member functions one being a static randomizer function which generates a random with number of digits passed as parameter. It also contains a pure virtual function Multiply which is overloaded in three other classes each corresponding to one algorithm all of which publicly inherit the Multiplicator class.



The three classes are namely GSA, Ktsba and DC, each one calls the virtual function Multiply and implements the multiplication algorithm within it.







Grade School Algorithm employs a brute force method with two loops, the first which fixes an element of number and multiplies it to elements of second number which is done in the loop and adds all results together after adding necessary amount of zeroes.

Karatsuba Algorithm employs recursion where every number is split into two and separated and again called in recursion until they are 1 digit each and can be directly multiplied (this is the base case).

Divide and Conquer Algorithm has a very similar approach as Karatsuba Algorithm, however it has a different formula as seen above.

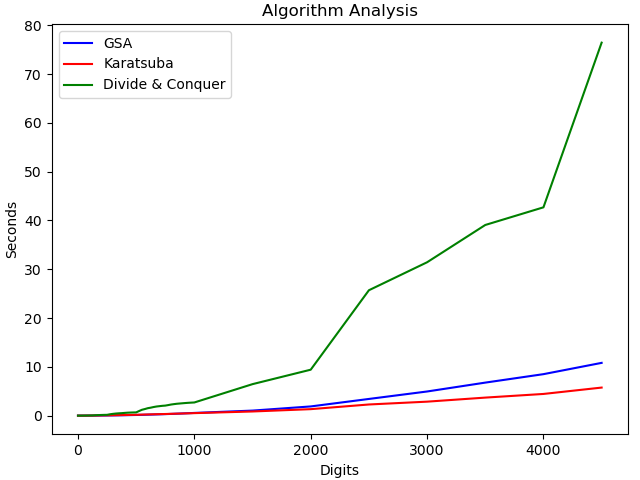
The cpp file Source.cpp calls Multiplicator.h header file and has a function JoinCreate that creates a shared pointer for Multiplicator class and creates shared pointer to class instances implementing these algorithms, opens a csv file and runs all the algorithms in a loop of specified time, 3 times each calculates average and stores them in the csv file.

A Python program has been implemented that reads this csv file using pandas library and appropriately makes graph using the matplotlib library.



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

Result

The following graph was constructed using a step of 25 until 1000 and till 5000 using 500 

From the dataset obtained it can be noted that up until 51 digits the run times for algorithms are notably similar, however after that it can be seen that Divide and Conquer Algorithm’s run time starts increasing steadily.





By around 500 digit mark it can be clearly seen in the graph that it is increasing. From the dataset it can be seen again that by 900 digit mark Karatsuba Algorithm begins to take less time than Grade School Algorithm.





However the difference can be truly seen in the graph at 2000 digit mark. By 2500 digit mark the time taken for Divide and Conquer has increased dramatically





The same can be seen by 4500 mark where again an even more significant increase can be seen





Conclusion

The analysis has been successfully carried out. The results are more or less in accordance with theoretical results, curve of Divide and Conquer as well as Grade School Multiplication has , however Divide and Conquer does work slower because of possible constant factor. Karatsuba Algorithm curve is also in accordance with results. If I were to improve my code, I would include more decompositions for finding average and creating csv as well as being able to connect the python graph program with Source.cpp.