Algorithms Design and Analysis

Visualization and Analysis of Closest-Pair and Karatsuba Algorithms

Ammar Hyder

Abstract

This report examines the Closest-Pair and Karatsuba algorithms, both of which use a divide-and-conquer approach to solve high computing problems. The Closest-Pair Algorithm identifies the closest points on a plane with a time complexity of O(n log n), an improved version of the brute-force $O(n^2)$ method. The Karatsuba Algorithm performs fast long integer multiplication with a complexity of $O(n^{1.585})$, than the traditional $O(n^2)$ for the brute force approach. A web-based visualizer was developed to demonstrate these algorithms, particularly for large datasets. Experimental results confirm the efficiency of the divide-and-conquer approach over brute force.

Introduction

In the field of mathematic use of geometry and multiplication is quite common, mathematician utilize formulas and complex techniques to map and obtain their findings and solutions, particularly when it comes to the above-mentioned topic of this discipline. Visualizing and finding closest pairs, if done manually, can be quite draining and exhaustive process, same goes for multiplication when it comes to extremely long integers, as it becomes simply impossible for a normal human being to perform all calculations.

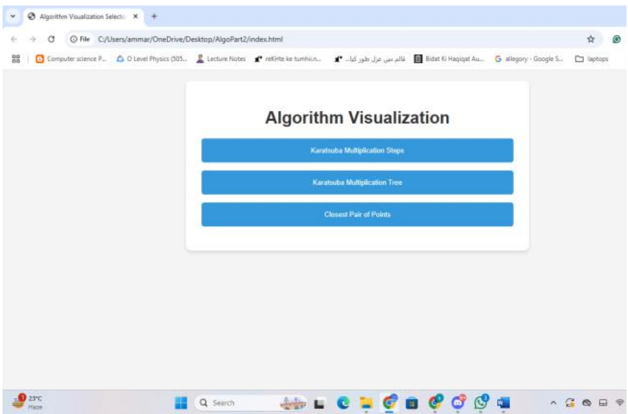
In order to solve these problems, algorithms were written and developed, to find the closest pair of points on a plane, the Closest-Pair Algorithm was developed, and to solve the long integer multiplication the Karatsuba Algorithm was developed. Both of these algorithms provided for a more easy-going computation. However, they came with their draw, for particularly large data sets the algorithms to too much time to compute and often times failed to produce results. Enhaced were made to these algorithms, adopting a divide and conquer approach rather than the slower brute force approach with respect to large data sets.

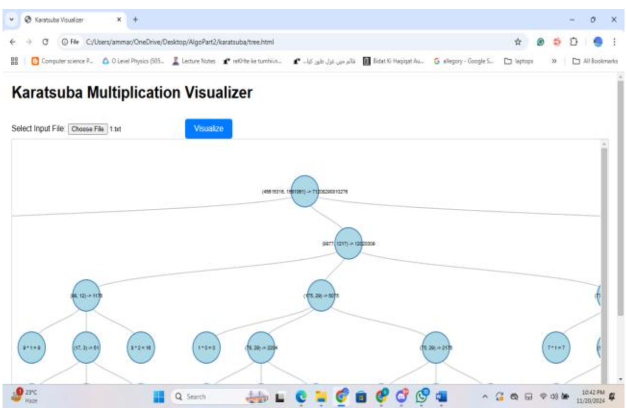
In this report we will be providing a solution for solving these mathematical problems, and in this report, we will be exploring as to how the brute force approach differs from the divide and conquer approach. The report provides a clear outline of the developed system, its working and discussion on the results generated by our proposed system.

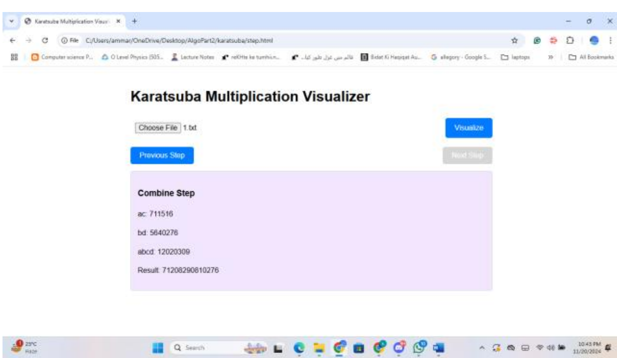
Proposed System

The system is a web-based Algorithms Visualizer that supports visualization of 2 Algorithms, Closest-Pair Algorithm and Karatsuba Algorithms, for Karatsuba Algorithms further 2 different variations have been made, one that shows mathematical steps performed to achieve the desired result, and the second that shows those steps in form of a Tree for better understanding of the algorithm's steps. Whereas for closest pair algorithms, a visualizer that shows the animation being performed when the points are being checked, the points are plotted on screen and lines are drawn between them when a pair is being checked.

Here is the diagram/UI of the system.







Both algorithms are based on the divide and conquer approach, resulting their time complexity to be **(n log n),** below is the explanation that shows as to why is this complexity.

Closet Pair

- Sorting point by x coordinate take **n log n** time as using JavaScript built-in method .sort() that uses mergesort.
- Recursive splitting of points into 2 parts take **log n** time.
- Brute force method only works on size 3, so constant time.
- Merging of the split result in **n** time. T(n)=2T(n/2)+O(n)

So overall dominant time complexity is

O(nlogn).

Application:

Air traffic monitoring making planes as point, then avoiding collisions.

Karatsuba

- Splitting the input numbers into smaller parts involves string manipulation and padding. This operation works in linear time **n.**
- Recursively splitting into halves takes **log n** time.
- Multiplication of integers at base case take constant time.
- Combining each recursion level takes **n** time.

T(n)= 3T(n/2)+O(n)

Apply Master theorem to get.

The overall time complexity is **O($n^{1.585}$).**

Application:

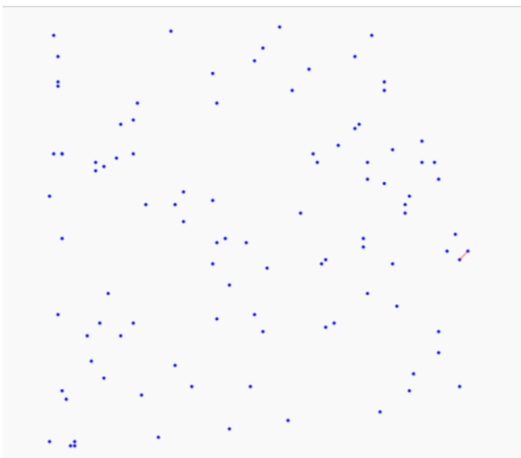
Multiplication of public and private keys to generate encryption in RSA.

Experimental Setup

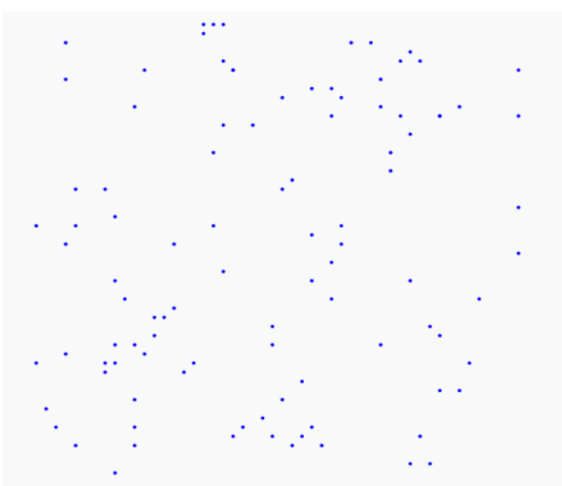
To test and check both of the algorithms, datasets have been generated contain random data of varying time complexities and cases. Each algorithms have 10 datasets on which the code will run and test.

Closest-Pair

Data Set 1(Close together)



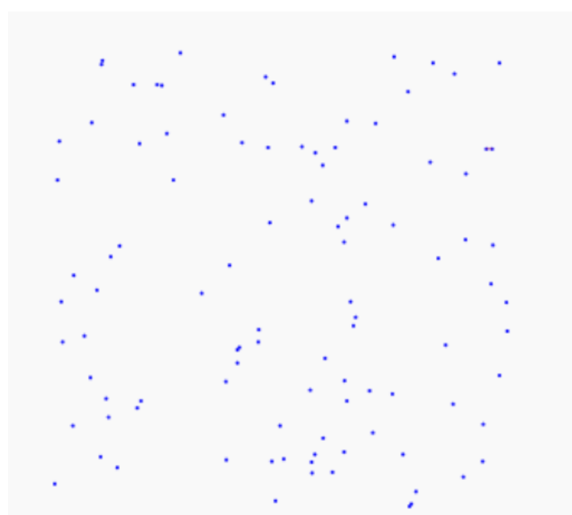
Data Set 2(Clustered)



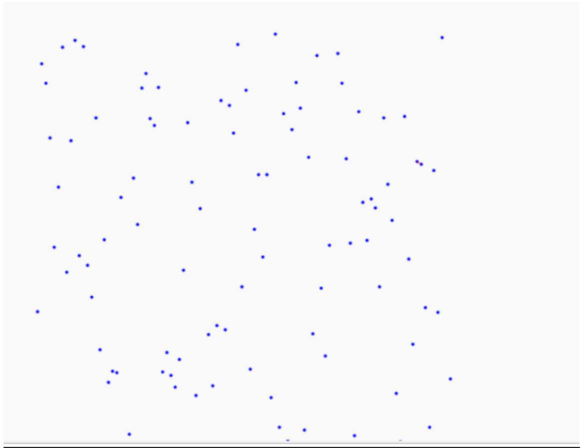
Data Set 3(Duplicates)

Data Set 4(Grid)

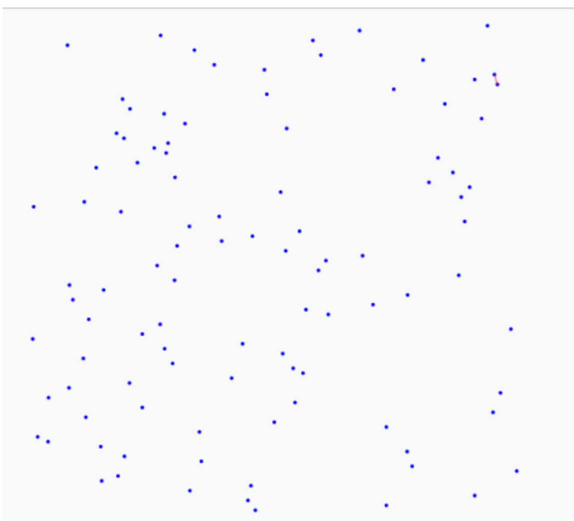
Data Set 5(Random)



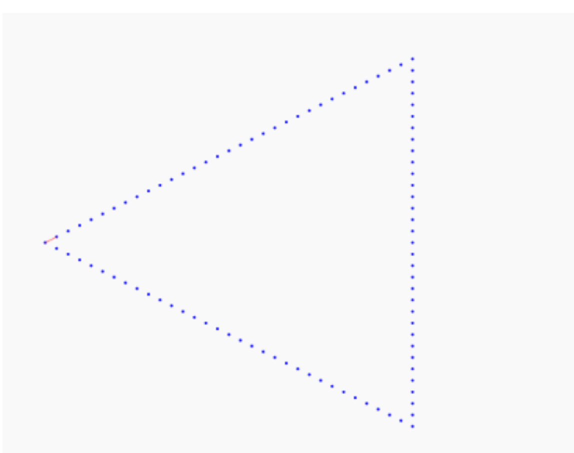
Data Set 6(sorted by x)



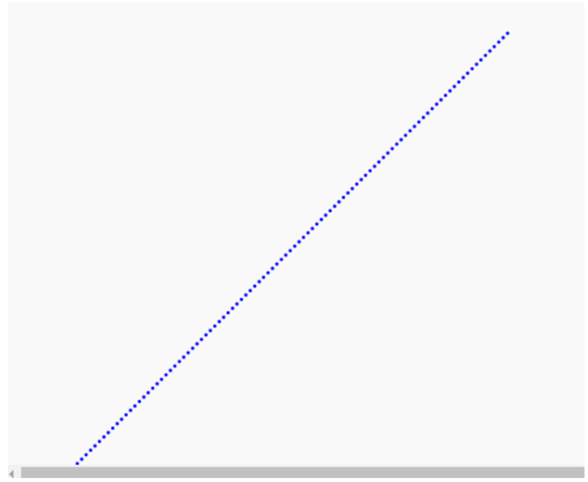
Data Set 7 (sorted by y)



Data Set 8 (geometric triangle)



Data Set 9(single line positive grad)

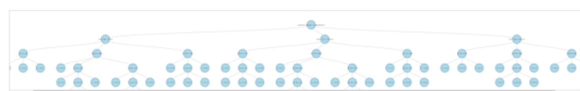


Data Set 10

Karatsuba

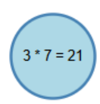
File1

Different sized integers



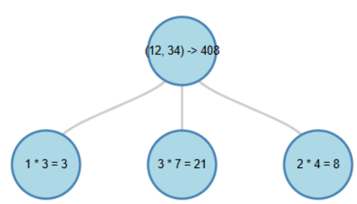
File2

Single digit integer



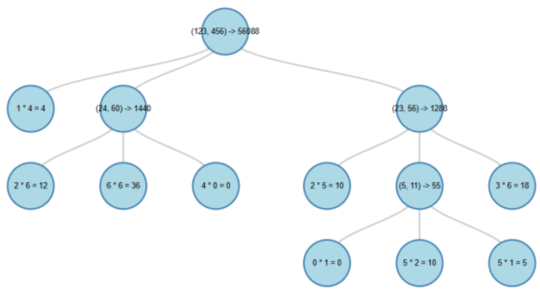
File3

Double digit integers



File4

Three-digit integers

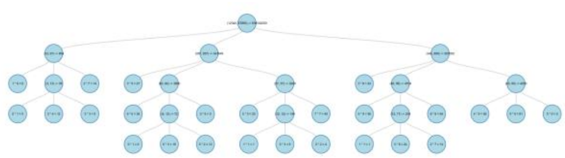


File5

Four-digit integers

File6

Five-digit integers

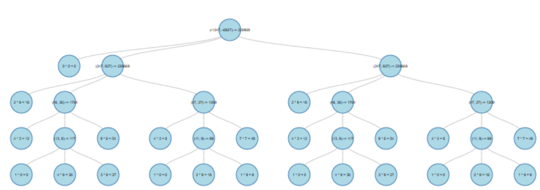


File7

Six-digit integers

File8

Both Negative integers



File9

One Negative one positive integer

File10

One zero, one regular integer

Results And Discussion

Karatsuba

Results/output shown in previous section. The results shown have been generated using the divide and conquer approach resulting in on an avg $n^{1.585}$ time, had this been done using the brute force approach the time taken would have n^2 far greater, and would have resulted in slower generation of the results.

Closest Pair

The algorithm was used on the datasets using divide and conquer approach resulting in a time complexity of n log n instead of n^2 had brute force approach been used.

If we look for an avg size of dataset, that is 100, the time would have 100*100 = 10000 units for the brute force, but for divide and conquer 100log100 = .664 (approx.) This clearly shows that the performance and time for the divide and conquer is far better than the brute force approach.

The results to both problems were checked against using algorithms for both from credible website, and the results matched, ensuring and reenforcing the results produced by the system developed.

Conclusion

This report has proved usefulness in use of divide and conquer in solving mathematical problems such as; the algorithm for finding the two closest points and long integer multiplications using the Algorithms Closest-Pair and the Karatsuba respectively. Using the developed system for the algorithms visualization, it was able to demonstrate significant savings in the time complexity as related to brute force approaches. The experimental results corroborated that for larger datasets, the time costs on computation with the divide-and-conquer strategy were much more favorable making it a more appealing approach

References

- IMPLEMENTATION OF KARATSUBA ALGORITHM USING POLYNOMIAL MULTIPLICATION
- Closest Pair Geeks for Geeks