REPORT

a. Abstract

This project implements two classical Divide & Conquer algorithms, **Closest Pair of Points** and **Karatsuba Multiplication** (for multiplying large integers 100–150 digits), through a Python-based system that generates datasets, parses input text files, executes the algorithms, computes results, and presents an interactive GUI for visualization.

b. Introduction

Divide & Conquer is a fundamental algorithmic approach used to solve complex computational tasks by recursively breaking them into smaller subproblems. This reduces computational complexity and improves performance in tasks that involve large datasets or heavy arithmetic operations. In this project, we implemented two Divide & Conquer algorithms:

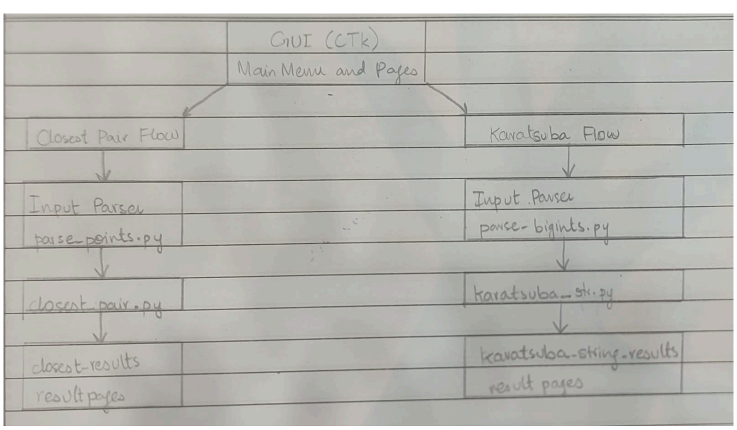
1. **Closest Pair of Points** – Computes the minimum Euclidean distance among *n* points in a plane in *O(n log n)* time.
2. **Karatsuba Multiplication** – Performs fast multiplication of large integers using a recursive method, improving from $O(n^2)$ to *O(n)* time.

We built a GUI using **CustomTkinter** to give the user an interactive environment to select input files, visualize datasets, and see computed outputs.

c. Proposed System

The diagram shows the full workflow of the system: the GUI lets the user choose an algorithm, after which the selected input file is read through its parser. The parsed data is then processed using either the Closest Pair or Karatsuba divide-and-conquer algorithm. Finally, the computed results are displayed back to the user.

System Diagram



d. Experimental Setup

Datasets for Closest Pair

The script generates **10 datasets**, containing **100–200 randomly generated 2D points**, each within the range 0–10,000.

Datasets for Karatsuba Multiplication

The generator creates **10 datasets** of integer pairs with **100–150 digits** each.

e. Results and Discussion

Closest Pair of Points

For all 10 datasets, the divide-and-conquer algorithm returned the correct minimum distances. Typical results:

- Distances varied between **30 and 300,** depending on the point distribution.
- The algorithm achieved expected *O(n log n)* behavior.
- When large datasets were passed via the GUI, the result appeared instantly, confirming efficient recursion.

Karatsuba Multiplication

- Karatsuba successfully multiplied all **100–150 digit** numbers
- Output numbers reached **200–300 digits**.
- Karatsuba's divide-reduce approach significantly improved performance.
- Outputs are displayed instantly in the GUI.

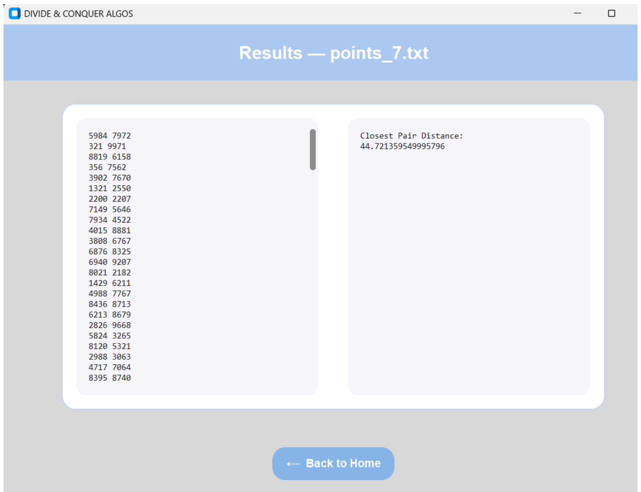
GUI Output Validation

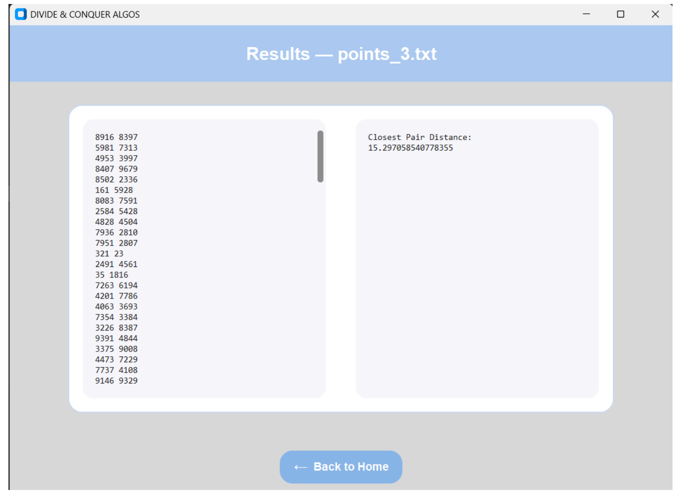
The GUI shows:

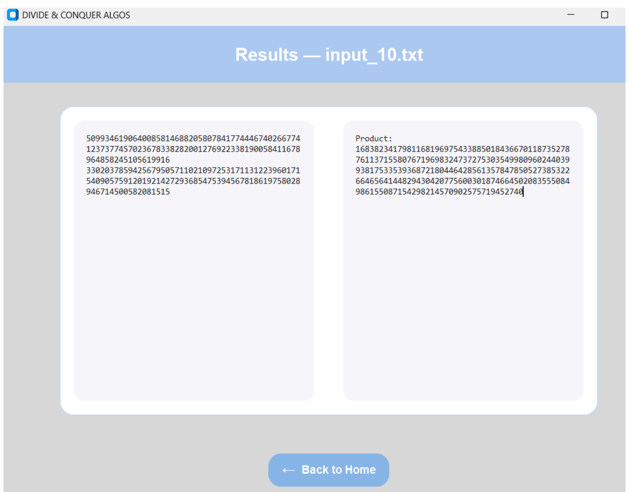
1. Raw input file contents
2. Computed result (distance or product)

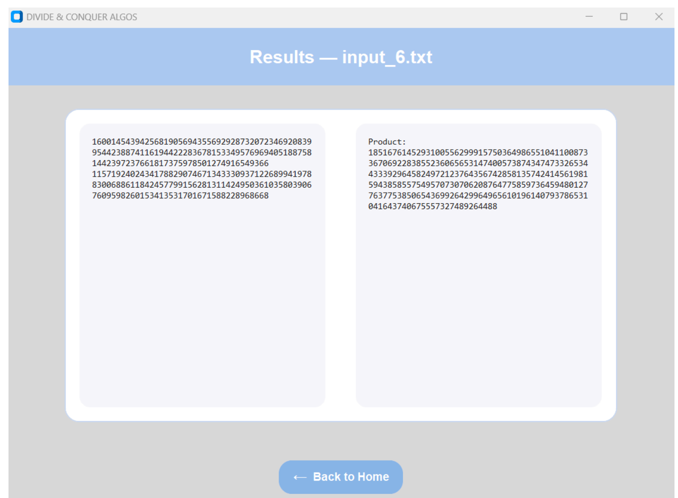
Side-by-side layout makes validation easy and visually clear.

Examples:









f. Conclusion

This project demonstrates the effective use of Divide & Conquer through the Closest Pair and Karatsuba Multiplication algorithms. With automated random dataset generation, parsing, recursive processing, and a Python interface, the system confirms both the correctness and efficiency of these methods. Overall, the work highlights the advantages of divide-and-conquer for geometric computations and large-integer arithmetic within a clear and modular design.

g. References

1. T. H. Cormen et al., *Introduction to Algorithms*, MIT Press.
2. Gustavson, Karatsuba's Algorithm (1962) — Fast multiplication research.
3. Plane geometry & Divide-and-Conquer literature.
4. Algorithm Design, Jon Kleinberg, 1st Edition
5. Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition