NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

(KARACHI CAMPUS)

FAST School of Computing

FALL 2024

PROJECT REPORT

DESIGN AND ANALYSIS OF ALGORITHMS

PROJECT MEMBERS:
22k-4413 Syeda Fakhira Saghir
22k-4448 Aafreen Mughal
22k-4499 Muhammad Raza

REPORT SUBMITTED TO: Sir Abu Zuhran Qaiser

Abstract
Commenced in October, 2024. This project focuses on solving two algorithmic problems using divide-and-conquer techniques: finding the closest pair of points in a 2D plane and performing integer multiplication. These problems were solved using randomized input datasets, with the results tested for correctness and efficiency. The divide-and-conquer approach allows for an efficient solution to the closest pair of points problem with a time complexity of O(n log n), and a fast method for integer multiplication based on Karatsuba's algorithm. The user interface provides a seamless experience, allowing users to select input files and see the results of the algorithms.

TABLE OF CONTENTS

A. INTRODUCTION

The divide-and-conquer technique is widely used in algorithm design. In this approach a problem is divided into smaller subproblems that are solved independently, and then the solutions to the subproblems are combined to solve the original problem. This project focuses on implementing two divide-and-conquer algorithms: finding the closest pair of points and integer multiplication. The closest pair of points problem is a classical problem in computational geometry, while integer multiplication using Karatsuba's algorithm provides a more efficient approach compared to traditional multiplication methods. Both problems were tested with randomized large input of varying sizes to evaluate the performance of the algorithms.

B. PROPOSED SYSTEM

Closest Pair of Points Algorithm:

- **Objective**: Efficiently find the smallest distance between two points in a 2D plane.
- **Approach**:
    - Sort points by x-coordinate.
    - Recursively divide the dataset into left and right halves.
    - Use a "strip" approach to compute distances between points across the partition line.
- **Complexity**: O(nlogn).

Integer Multiplication (Karatsuba's Algorithm):

- **Objective**: Efficiently multiply two large integers.
- **Approach**:
    - Split each number into halves.
    - Multiply recursively using the formula:

$$x \cdot y = 10^m \cdot a \cdot c + 10^{m/2} \cdot (a \cdot d + b \cdot c) + b \cdot d$$

    - Reuse partial results to reduce the number of multiplications.
- **Complexity**: O($n^{log_2 3}$) where ($n^{log_2 3}$) ≈ 1.585

C. EXPERIMENTAL SETUP

- Both algorithms are implemented with file handling for input and output.
- A GUI interface allows easy file selection and displays results.
- Efficient recursive structures are used to improve runtime.

Closest Pair of Points

  - Generates 10 files, each containing 101–250 randomly generated points with coordinates in the range [0, 1000).
  - Allows users to select a file to process.
  - Outputs the closest pair and their distance.
  - Closest Pair of Points: Randomly generated integer coordinates (x, y) for ranging between 100 to 150 points.
- **Features:**
  - Implements brute-force and divide-and-conquer approaches.
  - Uses recursive structure to split the dataset into halves.
  - Efficiently narrows the search space using a strip method sorted by the y-coordinate.
- **File Handling:**
  - Files contain random 2D points.
  - Allows users to select files to process.
- **Integrates a GUI to:**
  - Display results (smallest distance and closest pair) in a separate window.
  - Uses custom CSS for styling GTK+ buttons.

Integer Multiplication

The code generates 10 files containing random integers ranging from 10 to 200. It allows users to select two files for multiplication and displays results. The results are also saved in an output file.

- **Features:**
  - Implements Karatsuba's algorithm with recursive splitting.
  - Files generated contain random integers
- **File Handling:**
  - Reads input files line by line, converting strings to integers, and uses them in multiplication.
- **Provides GUI functionalities:**

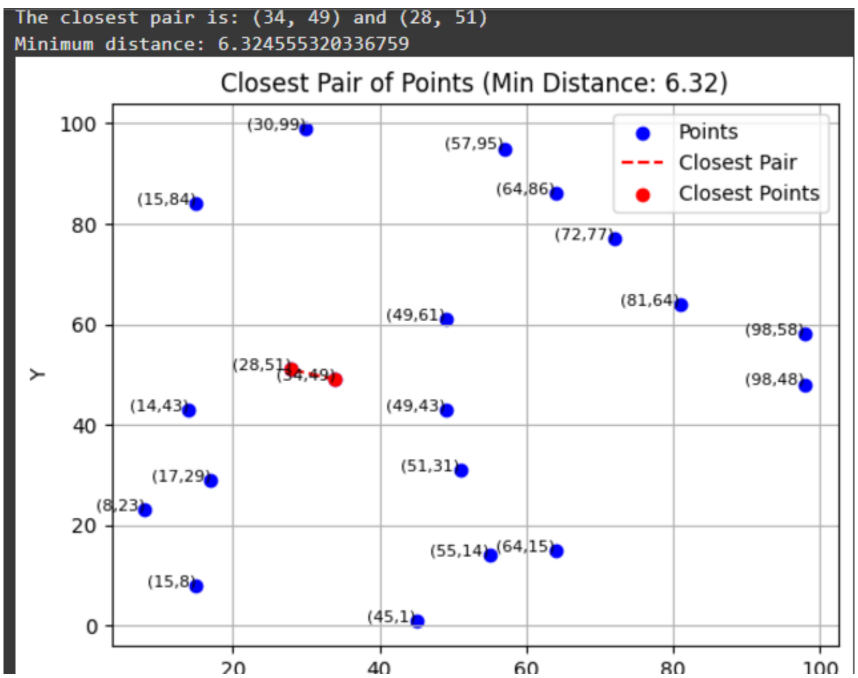
   - Display created filenames.
   - Input file names and compute results.
   - Outputs the results to an external file (OUTPUTQ2(ii)).

C. RESULTS AND DISCUSSION

Program Efficiency:

Closest Pair of Points:

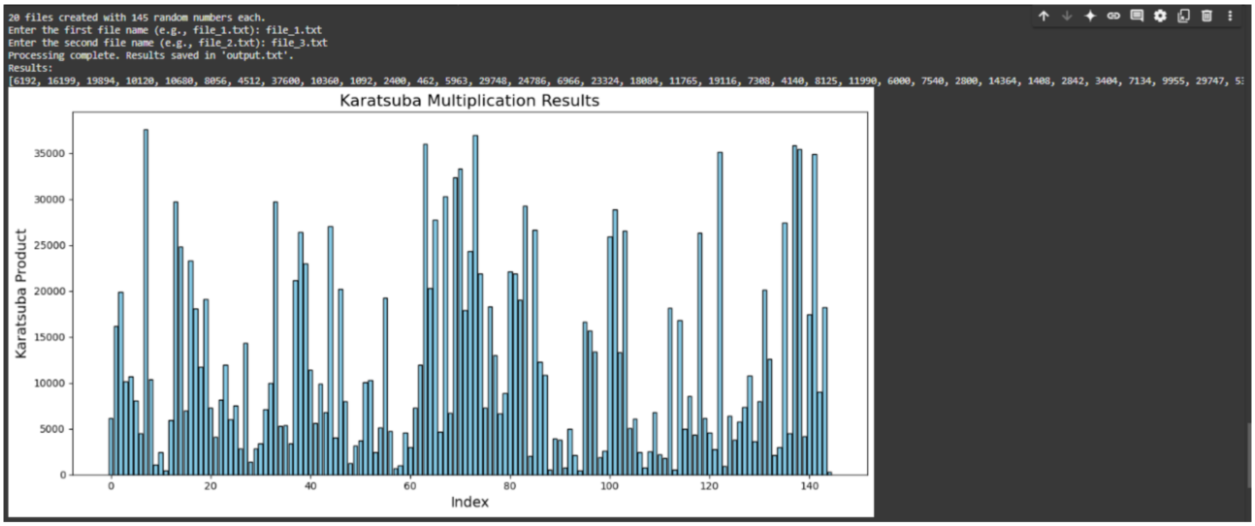
The graph below displays points plotted in a 2D space. The closest pair of points are highlighted in red based on their Euclidean distance. This illustrates the implementation of a closest-pair algorithm (e.g., divide-and-conquer) used in computational geometry to find two points with the smallest distance among a set.



- Complexity:
   - Sorting: $O(n\log n)$
   - Divide-and-conquer: $O(n\log n)$
   - Overall: $O(n\log n)$, which is efficient for moderate input sizes.
- Suggestions for Optimization:
   - For extremely large datasets, consider parallelizing the recursive calls using multithreading or GPU computing.
   - Use pre-sorted data (if available) to skip initial sorting.

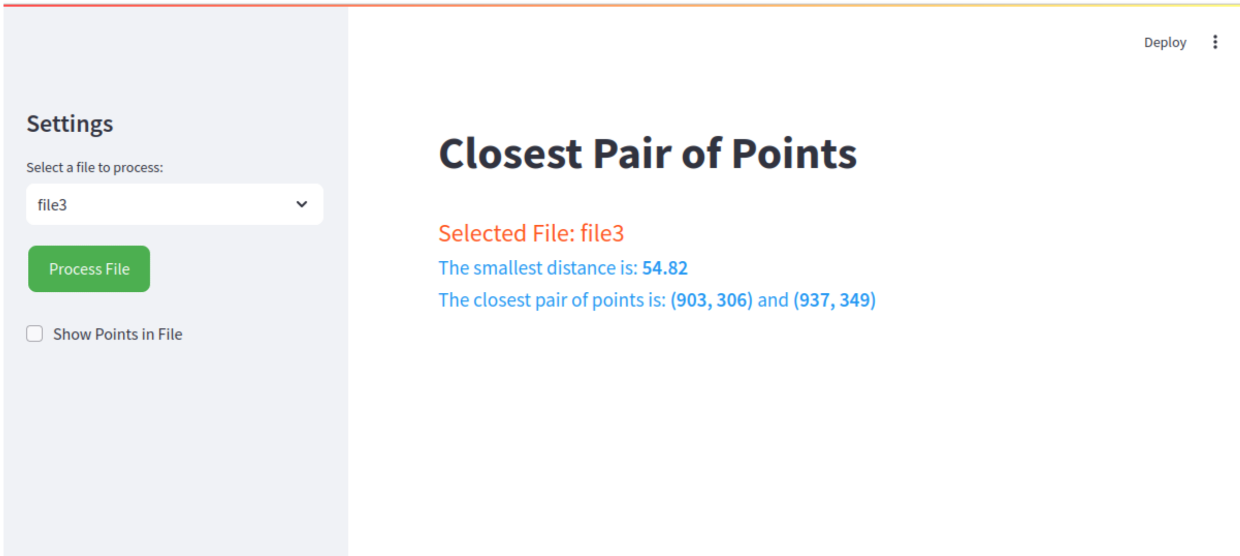
Karatsuba's Algorithm

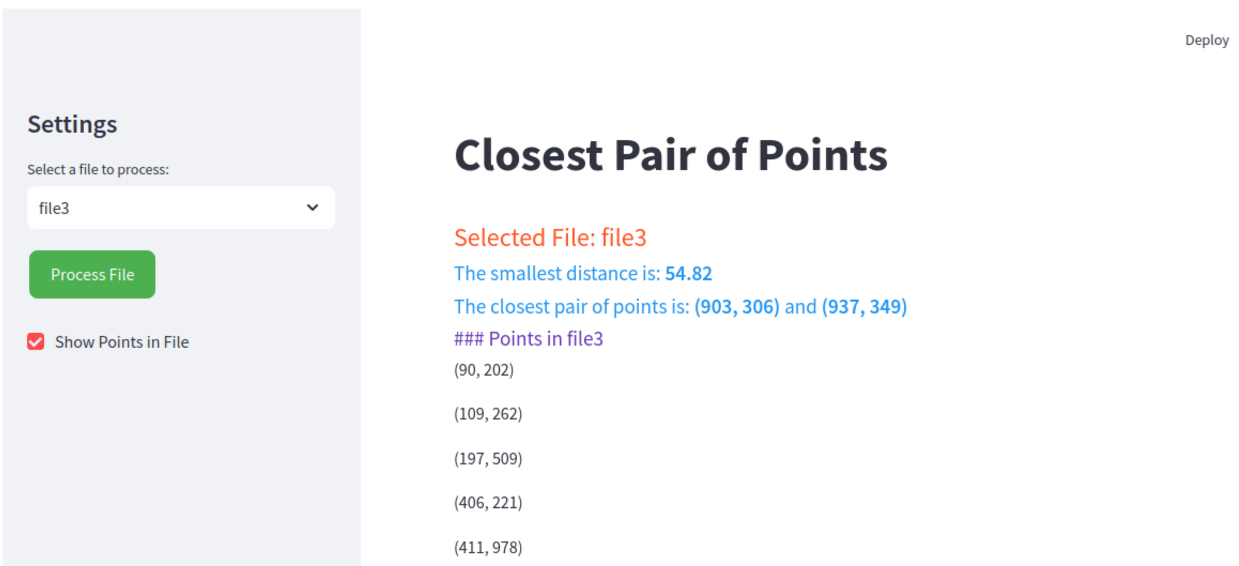
The bar chart visualizes the results of Karatsuba multiplication across various indices. Karatsuba is an efficient algorithm designed for multiplying large numbers quickly. Each bar represents the product of a specific calculation, with the corresponding values listed above. This chart displays how the algorithm performs on a randomized dataset and also gives an insight into the distribution of the results.



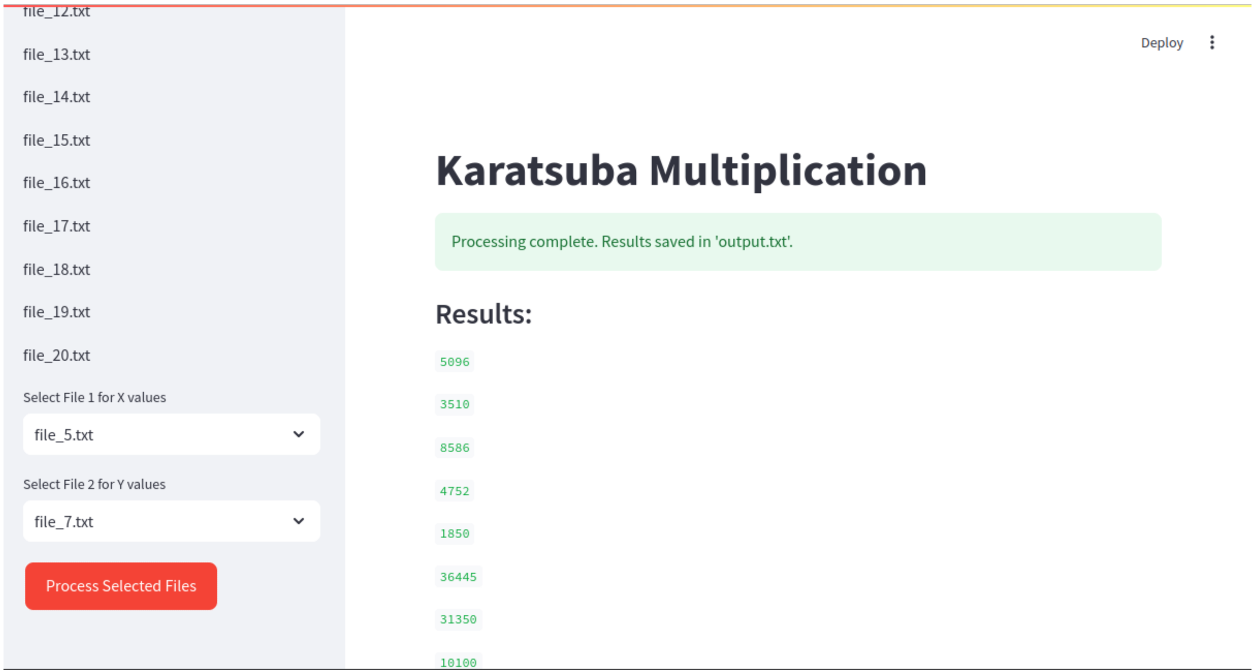
- Complexity:
  - Recursive splitting: O($n^{\log_2 3}$), which is faster than O(n^2) for large n.
- Suggestions for Optimization:
  - For very large integers, hybridize Karatsuba with FFT (Fast Fourier Transform) for further speedup.
  - Use memoization to store intermediate results if the same inputs are multiplied repeatedly.

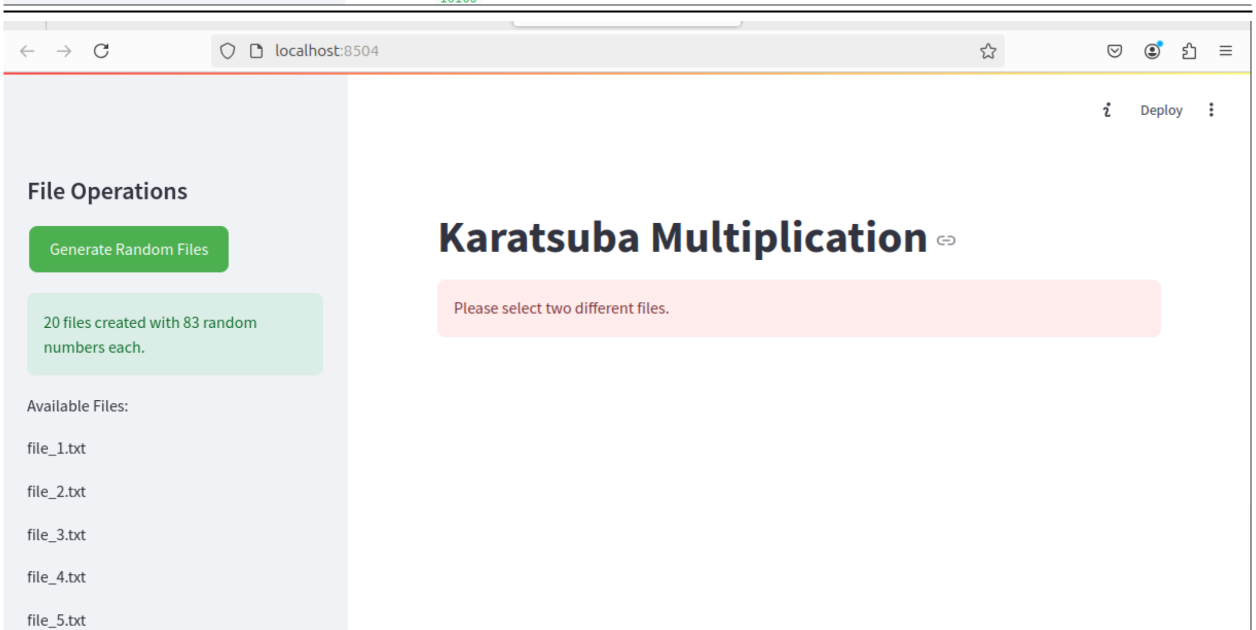
Q4 i


Deploy
Settings
Select a file to process:
file3
Process File
Show Points in File
Closest Pair of Points
Selected File: file3
The smallest distance is: 54.82
The closest pair of points is: (903, 306) and (937, 349)


Deploy
Settings
Select a file to process:
file3
Process File
Show Points in File
Closest Pair of Points
Selected File: file3
The smallest distance is: 54.82
The closest pair of points is: (903, 306) and (937, 349)
Points in file3
(90, 202)
(109, 262)
(197, 509)
(406, 221)
(411, 978)

Q4 ii





D. CONCLUSION

The project successfully implemented two divide-and-conquer algorithms with a user-friendly GUI. The Closest Pair of Points algorithm showcased efficient computation for 2D geometry problems, while Karatsuba's Algorithm highlighted the power of recursive approaches for integer multiplication. These implementations demonstrate the practicality of divide-and-conquer methods in solving computationally intensive tasks.

E. REFERENCES

1. https://www.geeksforgeeks.org/karatsuba-algorithm-for-fast-multiplication-using-divide-and-conquer-algorithm/
2. https://www.geeksforgeeks.org/closest-pair-of-points-using-divide-and-conquer-algorithm/
3. Textbook on Algorithms: Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. *Introduction to Algorithms.* MIT Press
4. GTK Documentation: https://www.gtk.org/
5. https://brilliant.org/wiki/karatsuba-algorithm/