Introduction

This report presents the implementation and comparison of various Divide and Conquer and Greedy algorithms using Object-Oriented Programming (OOP) concepts in Java.

Divide and Conquer Algorithms

QuickSort: Implements the QuickSort algorithm for sorting an array of integers.

MergeSort: Implements the MergeSort algorithm for sorting an array of integers.

Closest-Pair Problem: Finds the closest pair of points in a 2D plane.

Strassen's Matrix Multiplication: Efficiently multiplies two matrices.

Quickhull: Finds the convex hull of a set of points in 2D.

Greedy Algorithms

Prim's Minimum Spanning Tree (MST): Finds the minimum spanning tree of a graph.

Traveling Salesman Problem (TSP) (Approximate Solution): Finds an approximate solution for the TSP.

Kruskal's MST: Finds the minimum spanning tree of a graph using Kruskal's algorithm.

Dijkstra's Shortest Path: Finds the shortest path from a source node to all other nodes in a graph.

Huffman Codes: Constructs a Huffman tree and generates Huffman codes for given frequencies.

Design and Implementation

The application is designed using OOP concepts, ensuring modularity, readability, and maintainability. Abstract classes and interfaces are used to generalize algorithm behaviors, and inheritance and polymorphism are demonstrated in the design.

Performance Comparison

The application measures the execution time of each algorithm, allowing users to compare the performance of different algorithms on the same input. The results show that Divide and Conquer algorithms are generally more efficient for sorting and matrix multiplication, while Greedy algorithms are effective for graph-related problems.

Conclusion

This project successfully implements and compares various Divide and Conquer and Greedy algorithms, demonstrating a strong understanding of OOP concepts and algorithm design in Java.