Introduction

Sorting is a fundamental operation in computer science, forming the foundation for searching, indexing, and numerous algorithmic optimizations. Many sorting algorithms have been developed, each with distinct time complexities, memory requirements, and stability guarantees. In previous coursework, I implemented heapsort and quicksort and studied self-balancing binary search trees, which form the basis for tree sort. Merge sort is another classical divide-and-conquer algorithm widely used in real systems. For this project, a fifth algorithm—block sort—was included as a modern, theoretically efficient variation of merge sort that aims to reduce auxiliary space usage.

This paper presents an overview of each algorithm, accompanied by a visualization of its core behavior on a representative input array. A full experiment is then conducted to measure performance across the five methods. The remainder of the paper is organized as follows: five sections describe the algorithms, a methodology section outlines the experimental design, results summarize the empirical findings, and conclusions reflect on the outcomes and suggest areas for further study.