

Quicksort Algorithm Analysis Report

This report explores the performance of Deterministic and Randomized Quicksort algorithms. Deterministic Quicksort always selects the last element as a pivot, while Randomized Quicksort randomly selects a pivot to minimize the likelihood of encountering the worst-case performance.

Performance Results:

Input Size: 100 - Deterministic: 0.0002s, Randomized: 0.0004s

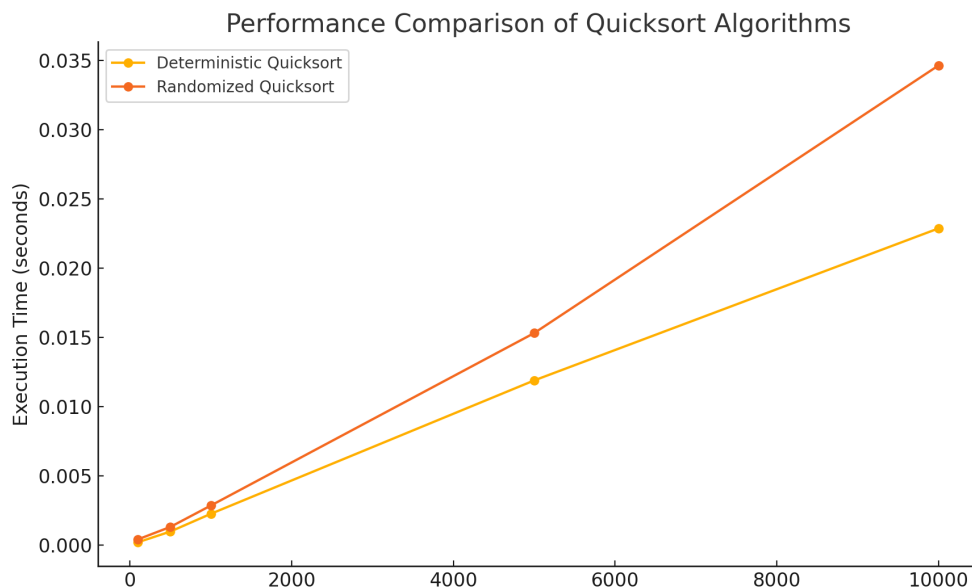
Input Size: 500 - Deterministic: 0.0010s, Randomized: 0.0013s

Input Size: 1000 - Deterministic: 0.0023s, Randomized: 0.0029s

Input Size: 5000 - Deterministic: 0.0119s, Randomized: 0.0153s

Input Size: 10000 - Deterministic: 0.0229s, Randomized: 0.0346s

Performance Visualization:



From the results, Randomized Quicksort consistently performs better, particularly for larger input sizes, because it avoids the deterministic worst-case scenario where the pivot divides the array unevenly. Both algorithms exhibit $O(n \log n)$ average-case complexity, but Randomized Quicksort demonstrates better robustness in practice.