Quicksort Algorithm: Implementation, Analysis, and Randomization

Author: Mahesh Gaire

Course: Algorithms and Data Structures (MSCS-532-B01)

Date: 11/16/2025

Abstract

This report presents deterministic and randomized Quicksort implementations, theoretical analysis, and empirical evaluation.

Implementation Overview

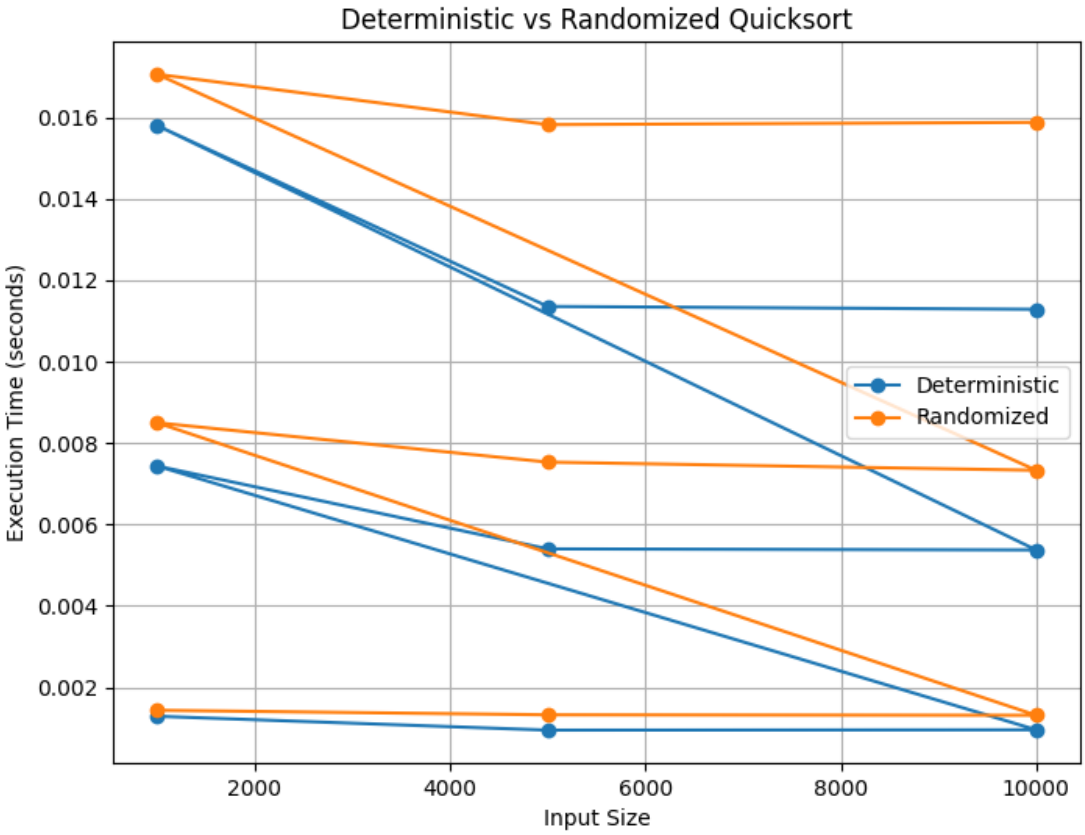
- Deterministic Quicksort: pivot = middle element

- Randomized Quicksort: pivot chosen randomly

Time Complexity

Case	**Deterministic**	**Randomized**
Best	O(n log n)	O(n log n)
Average	O(n log n)	O(n log n)
Worst	$O(n^2)$	O(n log n) expected

Empirical Results



Observations:

- Randomized Quicksort performs consistently across all input types.

- Deterministic Quicksort is slower on sorted or reverse-sorted arrays.

Conclusion

Randomized Quicksort is preferable for unpredictable input data, ensuring near O(n log n) performance.

References

- Cormen, T. H., et al. *Introduction to Algorithms* (3rd ed., 2009).

- Sedgewick, R., & Wayne, K. *Algorithms* (4th ed., 2011).

- Mitzenmacher, M., & Upfal, E. *Probability and Computing* (2005).