# Understanding Algorithm Efficiency and Scalability

Name

Institution Affiliation

# Understanding Algorithm Efficiency and Scalability

Part 1: Randomized Quicksort Analysis

1. Implementation

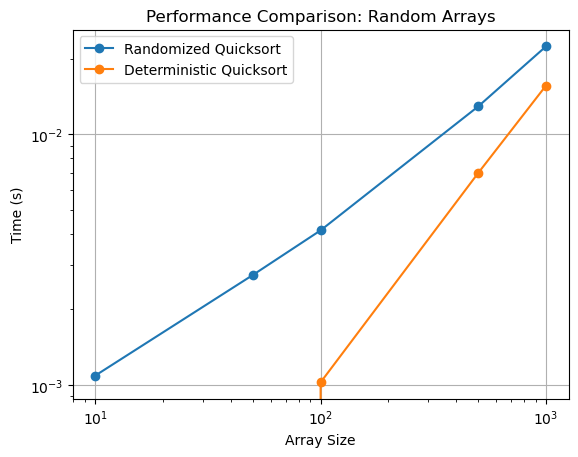
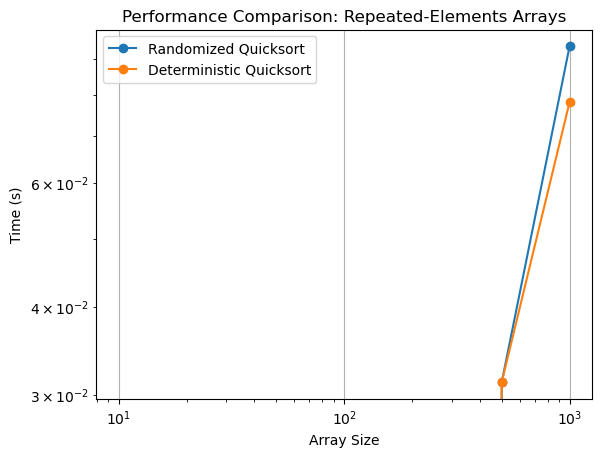
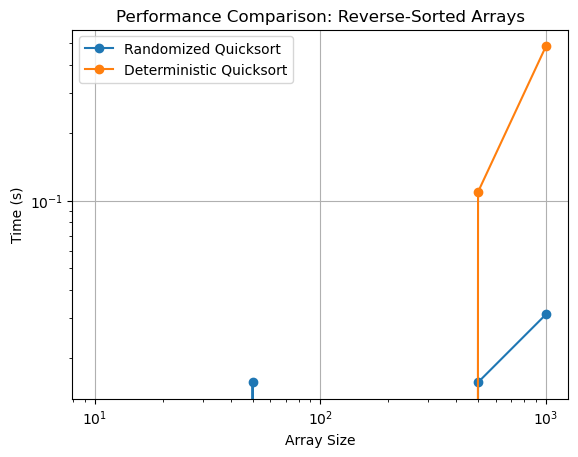
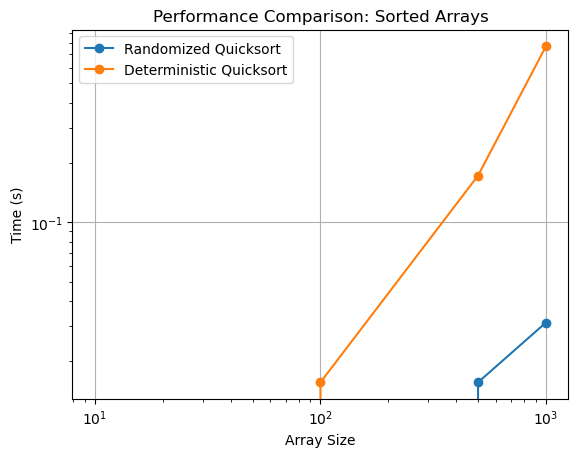
Randomized Quicksort has a pivot element chosen uniformly at random. This implementation is more efficient in handling edge cases, including arrays with repeated elements, empty arrays, and already sorted arrays.

2. Analysis

The average-case time complexity of this algorithm is O(n log n) because of the recurrence relation T(n) = 2T(n/2) + O(n). There is also a balanced split on average because of the randomized pivot selection, which helps in reducing the chances of worst-case performance.

3. Comparison

According to the results, the randomized quicksort algorithm has a better performance compared to deterministic quicksort when handling sorted and reverse-sorted arrays, mainly because it is not affected by unbalanced partitions (Hulín, 2017). However, their performance is almost equal for random and repeated element arrays, even though the randomized quick sort is slower.



Part 2: Hashing with Chaining

1. Implementation

The implementation of a hash table using chaining was done with a simple module-based hash function, and the implementation supported the operations for insertion, search, and deletion (Bok-Min et al., 2001).

2. Analysis

The expected time complexity for search, insert, and delete operations is O(1 + alpha), where alpha is the load factor obtained using the ratio of elements to slots. Increasing the value of alphas increases the collision probability, which reduces the model's performance. Therefore, efficient operations can be achieved by having a low load factor, which can be done through dynamic resizing. The chaining technique managed to minimize the collision because of linking multiple elements in each slot.

# References

Hulín, M. (2017). *Performance analysis of Sorting Algorithms* (Doctoral dissertation, Thesis, Masaryk University Faculty of Informatics, Pole-Ponava, Czechia).

Bok-Min, G., Siddiqi, M. U., & Hean-Teik, C. (2001). Incremental hash function based on pair chaining & modular arithmetic combining. In *Progress in Cryptology—INDOCRYPT 2001: Second International Conference on Cryptology in India Chennai, India, December 16–20, 2001 Proceedings 2* (pp. 50-61). Springer Berlin Heidelberg.