2024 4th Int. Conf. on Innovations in Science, Engineering and Technology (ICISET)
26-27 October 2024, Chittagong, Bangladesh

An Experimental Analysis on Different Pivot Selection Approaches for the Quicksort Algorithm

Jeba Tahsin
Department of Computer Science
American International University -Bangladesh
Dhaka, Bangladesh
ai.lily775@gmail.com

Dipta Gomes
Department of Computer Science
American International University -Bangladesh
Dhaka, Bangladesh
diptagomes@aiub.edu

Md. Manzurul Hasan
Department of Computer Science
American International University -Bangladesh
Dhaka, Bangladesh
manzurul@aiub.edu

***Abstract*—The research primarily examines the significance of pivot selection of the widely used QuickSort algorithm in order to increase the overall performance and efficiency. Quicksort has an average time complexity of $O(nlogn)$, but its performance can degrade to $O(n^2)$ in the worst-case scenario, which occurs when a pivot element is chosen badly. This study focuses on the influence of different pivot selection techniques on the efficiency of the Quicksort algorithm through empirical evaluation. To determine which strategy works best for an individual data set and array size, different methods have been evaluated, aiming to choose a pivot that is in close proximity to the median of the sub-array, evaluating their efficiency and any drawbacks. In terms of efficiency, the Median of Seven (MO7) and Median of Three (MO3) exhibits the best results, where MO7 gives an execution time of 0.0112s and MOT of 0.0124s. A comparative decision criteria has also been proposed in this research in choosing the optimum approach among the best performing MOT and MO7, where MOT being simpler and MO7 being more efficient. These insights offer practical guidance for optimizing Quick Sort implementations in real-world scenarios, where its performance is paramount.**

***Index Terms*—Quicksort, Pivot selection, Sorting algorithms, Experimental analysis, Median of Three (MOT), Median of Five (MO5), Median of Seven (MO7), Median of Nine (MO9), Random selection, Time complexity; efficiency; worst-case scenario, Quicksort variations.**

I. Introduction

Sorting algorithms are fundamental to various data processing tasks which plays an important role in organizing vast datasets efficiently. Quicksort, renowned for its high efficiency and speed, employs a divide-and-conquer approach . The algorithm recursively divides an array based on a selected pivot element, arranging elements smaller than the pivot to its left and bigger elements to its right. Optimally, the pivot should partition the array into about equal sub-arrays, minimizing comparisons and achieving efficient sorting. Nevertheless, the efficiency of Quicksort depends on the quality of the pivot selection [1]. Choosing a pivot positioned at one of the outermost ends of the sub-array, such as the first or final element, might result in imbalanced partitions [2]. This can greatly escalate the number of comparisons needed and lead to a complexity of $O(n^2)$.

In this research, we investigate the efficacy of different pivot selection strategies in enhancing Quicksort algorithm performance. Through experimental analysis and statistical evaluation, we aim to identify the most efficient pivot selection approach for optimizing the runtime. By shedding light on the strengths and limitations of each technique, this research seeks to contribute valuable insights to the field of sorting algorithms and algorithmic optimization.

In the following sections, we discuss the background of the Quicksort algorithm, explore various pivot selection strategies, outline the experimental design and methodology employed, present the obtained data, analyze the results, and discuss avenues for further research and exploration.

II. Quicksort Algorithm: Literature and Background

A. Previous Studies

The book titled *Quicksort and Sorting Lower Bounds* by Blelloch [1] offers an in-depth exploration of Quicksort and theoretical lower bounds for sorting algorithms, providing valuable insights into algorithm design and optimization. Similarly, Sedgewick et al. [3] present a comprehensive analysis of sorting algorithms, including Quicksort and Mergesort, with detailed explanations, pseudo-code, and performance analysis, making it a key resource for learners at all levels. Cormen et al. [4], in their seminal work, provide rigorous theoretical and practical insights into sorting algorithms, including Quicksort, establishing it as a foundational reference in the field. Blelloch et al. [2] further discuss lower bounds for sorting algorithms, highlighting theoretical limitations and challenges in achieving optimal worst-case complexity.

Recent research has focused on improving Quicksort's efficiency. Sabir et al. [5] proposes a variant that reduces worst-case time complexity to $O(nlogn)$ by minimizing comparisons through a manual sort for small inputs and using the mean as the pivot. Aoun Aftab et al. [6] introduce a dynamic pivot selection technique, the SMS-Algorithm, which organizes input into provisional arrays $(PosArray, NegArray, andFreqArray)$ and achieves a complexity of $O(n + f(max + min))$ for distinct elements. Aumüller et al. [7] propose a Dual-pivot Quicksort algorithm, subdividing input into three parts and achieving a complexity of $1.8nlnn + O(n)$ by reducing key comparisons.

979-8-3503-5549-9/24/$31.00