

Combining Quick Sort and Merge Sort for Improved Average-Case Performance

Lucky Gupta and Apoorv Chaudhary

Students, Computer Science and Application

Sharda School of Engineering & Technology, Sharda University, Greater Noida

Abstract: *This paper proposes and describes QuickMerge, a hybrid sorting algorithm -- that incorporates a fast in-place partitioning strategy (which is implemented by the classic Quick Sort algorithm) and a stable, cache-friendly merging strategy (which is implemented by the efficient and widely known Merge Sort algorithm) -- to improve the average case performance on a wide range of practical inputs. We present the algorithm design and pseudo-code, provide formal analysis of complexity (time, space, stability) and present a conceptual and empirical (based on published experimental results as well as trends reported in the literature) comparison of the performance between QuickMerge and that of classical and modern sorting algorithms, namely Quick Sort, Merge Sort, Intro Sort, Tim Sort, Radix/Counting sorts and recent hybrid/adaptive sorting algorithm proposals. Results indicate that a carefully designed QuickMerge - a hybrid where the constituent of Quick Sort partitions is used, but the constituent of Merge Sort merges (also referred to as balanced runs) is combined with a merge stage for large sized partitions - is able to combine the low overhead of the Quick Sort algorithm on the unspecialized data with Merge Sorts better steadiness on the adversarial or partially ordered input to improve the average wall clock times in many realistic workloads.*

Keywords: Quick Sort, Merge Sort, Heap Sort, Intro Sort

