A Study of the Ska Sort Algorithm

Enda Healion, Master in Computer Science

University of Dublin, Trinity College, 2024

Supervisor: Dr. David Gregg

Sorting algorithms play a very important role in software engineering, especially in use in databases where keys must be in sorted order so that search and merge functions are run more efficiently. As a result, it is important to have efficient algorithms that can sort a large number of inputs. Radix sorting algorithms are best suited to large input sizes due to their non-comparative sorting technique. Because of their importance, radix sorting algorithms are very well studied.

One of the main slow downs of in-place radix sorting algorithms, such as American Flag Sort, is the read and write dependencies when swapping elements. The Ska Sort algorithm shows significant performance improvements over American Flag Sort from its change in element swapping strategy. Despite the performance improvements, there is a gap in the literature about Ska Sort.

This paper studies the effects on the number of elements swapped with Ska Sort's new element swapping strategy. Due to the heavy usage of C++ templates and lambda functions, the changes in algorithm from American Flag Sort to Ska Sort is difficult to see. This paper takes the original implementation and extracts the algorithmic steps of Ska Sort. The original implementation contains optimisations which have not been proven to improve the runtime performance of the algorithms. This paper breaks down and analyses different sections of the algorithm into chapters to see if the optimisations provide runtime benefit or if there are any different optimisations that can be checked for performance improvements. Finally, this paper takes the learnings from Ska Sort and applies it to integer spreadsort to see if similar runtime improvements can be found.