This research will focus on analysing important properties of different sorting algorithms.

Stable/Unstable? – will depend on machine environment and stability of sorting algorithm itself.

Best/Average/Worst scenarios?

What is sorting algorithms?

In computer science, a sorting algorithm is an algorithm that puts elements of a list into an order. The most frequently used orders are numerical order and lexicographical order, and either ascending or descending. Efficient sorting is important for optimizing the efficiency of other algorithms (such as search and merge algorithms) that require input data to be in sorted lists. Sorting is also often useful for canonicalizing data and for producing human-readable output.

History – when did sorting algorithms emerge? Who is founder of sorting algorithms? How did they evolve?

From the beginning of computing, the sorting problem has attracted a great deal of research, perhaps due to the complexity of solving it efficiently despite its simple, familiar statement. Among the authors of early sorting algorithms around 1951 was Betty Holberton, who worked on ENIAC and UNIVAC.[1][2] Bubble sort was analyzed as early as 1956.[3] Asymptotically optimal algorithms have been known since the mid-20th century - new algorithms are still being invented, with the widely used Timsort dating to 2002, and the library sort being first published in 2006.

Comparison sorting algorithms have a fundamental requirement of Ω(n log n) comparisons (some input sequences will require a multiple of n log n comparisons, where n is the number of elements in the array to be sorted). Algorithms not based on comparisons, such as counting sort, can have better performance.

Sorting algorithms are prevalent in introductory computer science classes, where the abundance of algorithms for the problem provides a gentle introduction to a variety of core algorithm concepts, such as big O notation, divide and conquer algorithms, data structures such as heaps and binary trees, randomized algorithms, best, worst and average case analysis, time–space tradeoffs, and upper and lower bounds.

Sorting small arrays optimally (in fewest comparisons and swaps) or fast (i.e. taking into account machine specific details) is still an open research problem, with solutions only known for very small arrays (<20 elements). Similarly optimal (by various definitions) sorting on a parallel machine is an open research topic.

Sorting algorithms in real world applications.

Sorting algorithms are very important in our everyday life. Starting with google search or reading comments on any website.

Mythology – application (PETKO), hardware of Mino PC.