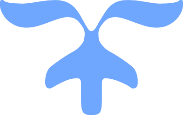


review on sorting algorithms

Computational Thinking with Algorithms

Lecturer: Dominic Carr



May 3, 2021

Higher diploma in science – computing (data analytics)

Olga Rozhdestvina

G00387844

Introduction

Introduce the concept of sorting and sorting algorithms, discuss the relevance of concepts such as complexity (time and space), performance, in-place sorting, stable sorting, comparator functions, comparison-based and non-comparison-based sorts, etc.

Sorting Algorithms (5 x 5 = 25%): Introduce each of your chosen algorithms in turn, discuss their space and time complexity, and explain how each algorithm works using your own bespoke diagrams and different example input instances. If you diagrams are not original creations you will get zero.

• Implementation & Benchmarking (25%): This section will describe the process followed when implementing the application above, and will present the results of your benchmarking. Discuss how the measured performance of the algorithms differed – were the results similar to what you would expect, given the time complexity of each chosen algorithm? In this section you should use both a table and a graph to summarise the results obtained (see samples below).

# Introduction

Sorting is a process of organising a list of elements in a particular order which makes handling the elements more efficient than handling randomize elements (S. Paira et al., 2014). Thus, sorting increased efficiency of each of the subsequent operations. This along with the rapid informational growth in our world lead to increase in the development of sorting algorithms.

Sorting algorithm is an algorithm that takes an array as input and outputs a [permutation](https://brilliant.org/wiki/permutations/) of that array that is sorted.

Developing sort algorithms through improved performance and decreasing complexity, it has attracted a great deal of research; because any effect of sorting algorithm enhancement of the current algorithms or product new algorithms that reflects to optimize other algorithms. Large number of algorithms developed to improve sorting like merge sort, bubble sort, insertion sort, quick sort ,selection sort and others, each of them has a different mechanism to reorder elements which increase the performance and efficiency of the practical applications and reduce time complexity of each one. When comparing between various sorting algorithms, there are several factors that must be taken in consideration; first of them is the time complexity, the time complexity of an algorithm determined the amount of time that can be taken by an algorithm to run [3][7][27]. This factor Khalid Suleiman Al-Kharabsheh, Ibrahim Mahmoud AlTurani, Abdallah Mahmoud Ibrahim AlTurani & Nabeel Imhammed Zanoon International Journal of Computer Science and Security (IJCSS), Volume (7) : Issue (3) : 2013 121 different from sorting algorithm to another according to the size of data that we want to reorder, some sorting algorithm inefficient and too slow. The time complexity of an algorithm is generally written in form big O(n) notation, where the O represents the complexity of the algorithm and a value n represent the number of elementary operations performed by the algorithm [8].The second factor is the stability[26], means; algorithm keeps elements with equal values in the same relative order in the output as they were in the input. [2][3][9]. Some sorting algorithms are stable by its nature such as insertion sort, merge sort, bubble sort, while some sorting algorithms are not, such as quick sort, any given sorting algorithm which is not stable can be modified to be stable [3]. The third factor is memory space, algorithm that used recursive techniques need more copies of sorting data that affect to memory space [3][9].Many previous researches have been suggested to enhance the sorting algorithm to maintain memory and improve efficiency. Most of these algorithms are used comparative operation between the oldest algorithm and the newest one to prove that.

The sorting algorithms examined in this report are:

# References

1. S. Paira, S. Chnadra, A. Sk Safikul, D.S. Partha (2014) ‘A Review Report on Divide and Conquer Sorting Algorithm’. Available at: <https://www.researchgate.net/publication/276847633_A_Review_Report_on_Divide_and_Conquer_Sorting_Algorithm> (Accessed: 3 May, 2021)