27/05/2019

CSP 2348 Assignment 2

A Mini Team Project

Samuel Morris – 10456767

Riegard Lemmer - 10424383

Sam

ECU JOONDALUP – CSP2348 – Data structures – assignment 2

# Executive Summary

Table of Contents

[Executive Summary 1](#_Toc9430874)

[Introduction 3](#_Toc9430875)

[Bubble and Heap Sort Algorithm Analysis 3](#_Toc9430876)

[Bubble sort algorithm analysis 3](#_Toc9430877)

[Heap sort algorithm analysis 3](#_Toc9430878)

[Summary of Sort Algorithm Complexities 4](#_Toc9430879)

[Algorithm Analysis by Experimental Studies 5](#_Toc9430880)

[Sequencing Array Sorting Algorithms Based on their Complexity 5](#_Toc9430881)

[Bubble Sort 5](#_Toc9430882)

[Selection Sort 5](#_Toc9430883)

[Insertion Sort 5](#_Toc9430884)

[. Merge Sort 5](#_Toc9430885)

# Introduction

# Bubble and Heap Sort Algorithm Analysis

## Bubble sort algorithm analysis

## Heap sort algorithm analysis

# Summary of Sort Algorithm Complexities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sorting Algorithm | No. of comparisons | No. of  Copies | Time complexity | Space complexity |
| Bubble | ~ O(n2) | n(n-1)/2 | BC: O(N)  WC: O(n2) | O (1) |
| Selection | *~* | *~ 2n* | *BC:* O(n2)  WC: O(n2) | *O (1)* |
| Insertion | *~* n2 */ 4* | *~*n2 */ 4* | *BC:O(N)*  *WC:* O(n2) | *O (1)* |
| Merge | *~n log2n* | *~ 2n log2n* | *BC: O (n log2n)*  WC:*O (n log2n)* | *O(n)* |
| Quick | *\*BC: ~n log2n*  *\*WC: ~* n2 */2* | *BC: ~2n/3 log2n*  *WC: 0* | *BC: O (n log2n)*  *WC:* O (n2)*)* | *\*BC: O ()*  *\*WC: O(n)* |
| Heap | O () | O () | BC: O ()  WC: O () | WC: O (n) |

https://www.geeksforgeeks.org/analysis-of-different-sorting-techniques/

# Algorithm Analysis by Experimental Studies

# Sequencing Array Sorting Algorithms Based on their Complexity

## Bubble Sort

The bubble sort algorithm has an average performance of O(n2), the performance of this algorithm is relative to its size, so It would rarely be used to sort large data sets. Bubble sort works well on small sets of data where the inefficiency doesn’t affect the overall performance as much. The most efficient use of bubble sort would be for a list that is mostly sorted as time would be relative to the number of unsorted elements. One unsorted element would give 2n time, two would take 3n time and so on.

## Selection Sort

The selection sort algorithm has O(n2) complexity, which is like the bubble sort as its performance is relative to its size. Selection sort is notable for how simple it is, and under specific circumstances has better performance than algorithms with greater complexity. It generally performs worse than much alike insertion sort.

## Insertion Sort

The insertion sort algorithm can be compared to an improved version of selection sort. It is quite efficient on smaller set of data and mostly sorted sets of data, because of this it is often used in conjunction with more complex algorithms. The downfall of insertion is that computationally expensive due to its shifting of all following elements when it shifts any element

## . Merge Sort

The merge sort algorithms worst run time is O (n log n), this means that it scales well when sorting large sets of data. Merge sorts efficiency stems from when

<https://resources.saylor.org/wwwresources/archived/site/wp-content/uploads/2011/06/Sorting-Algorithm.pdf>

**https://www.quora.com/Which-sorting-algorithm-is-best-and-why**