Analysis Report

for

Comparing Sorting Algorithms Project

Mustafa Talha Arslan (140144005)

mustafatalha.arslan@stu.thk.edu.tr

University of Turkish Aeronautical Association

Department of Computer Engineering

17 December 2016

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Properties of Test Environment 1

2. Test Results 1

3. Complexity Analysis 2

3.1 Time Complexities of Algorithms 2

3.2 Test Analysis 2

4. Test Graph 3

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Mustafa Talha Arslan | 17/12/16 | Initial work | 1 |

# Introduction

## Purpose

The purpose of this document is to analysis the test results of Insertion Sort, Quicksort with 2-way partitioning, Quicksort with 3-way partitioning and Bucket Sorting algorithms that are executed by using array sizes 10, 100, 1000, 10.000, 50.000 and 100.000.

## Properties of Test Environment

Toshiba Satellite P50-B-116 Notebook with Windows 10 Home Language

2.50 GHz 4th generation Intel® Core™ i7-4710HQ processor

16,384 (8,192 + 8,192) MB DDR3L RAM (1,600 MHz)

IDE: Dev-C++

# Test Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm\Array | 10 | 100 | 1000 | 10.000 | 50.000 | 100.000 |
| Insertion Sort | 0.014 s | 0.014 s | 0.015 s | 0.085 s | 1.803 s | 7.141 s |
| 2-way Quicksort | 0.014 s | 0.014 s | 0.013 s | 0.015 s | 0.019 s | 0.025 s |
| 3-way Quicksort | 0.013 s | 0.014 s | 0.014 s | 0.015 s | 0.019 s | 0.025 s |
| Bucked Sort | 0.014 s | 0.014 s | 0.014 s | 0.015 s | 0.018 s | 0.020 s |

The results of the test are held on the above chart after executions of all algorithms with very similar arrays. All algorithms executed as sets for each size of array to reduce error margin.

# Complexity Analysis

## Time Complexities of Algorithms

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Best | Average | Worst |
| Insertion Sort | Ω(n) | Θ(n^2) | O(n^2) |
| 2-way Quicksort | Ω(n log(n)) | Θ(n log(n)) | O(n^2) |
| 3-way Quicksort | Ω(n log(n)) | Θ(n log(n)) | O(n^2) |
| Bucked Sort | Ω(n+k) | Θ(n+k) | O(n^2) |

## Test Analysis

Test arrays was not worst case for sorting algorithms, so average performance of algorithms processed. Also, the test was done with very similar arrays and test environment was powerful. So, execution times do not make a huge difference for small array sizes such as 10, 100 and 1000. For bigger array sizes, such as 10.000, 50.000 and 100.00, insertion sort algorithm took a long time because insertion sort has worst average performance in the tested algorithms. After that, 2-way and 3-way partitioning quicksort algorithms are coming according to average performance but these algorithms have huge differences with insertion sort algorithm. Bucked sort algorithm has the best complexity in the tested algorithms, with small differences with quicksort algorithms.

In conclusion, Bucked Sort >> 2-way & 3-way Quicksort >> Insertion Sort, according to complexity.

# Test Result Graph

There is a graph according to Test Results Chart.