Lecturer: Dominic Carr

Computational Thinking With Algorithms

Student: Richard Deegan G00387896

Contents

[Preliminaries 2](#_Toc71897974)

[Introduction 3](#_Toc71897975)

[Sorting Algorithms 3](#_Toc71897976)

[Bubble Sort 3](#_Toc71897977)

[Introduction 3](#_Toc71897978)

[Space and time complexity 3](#_Toc71897979)

[How Bubble Sort works with original diagram 3](#_Toc71897980)

[Merge Sort 3](#_Toc71897981)

[Introduction 3](#_Toc71897982)

[Space and time complexity 3](#_Toc71897983)

[How Merge Sort works with original diagram 3](#_Toc71897984)

[Counting Sort 3](#_Toc71897985)

[Introduction 3](#_Toc71897986)

[Space and time complexity 3](#_Toc71897987)

[How Counting Sort works with original diagram 3](#_Toc71897988)

[Selection Sort 3](#_Toc71897989)

[Introduction 3](#_Toc71897990)

[Space and time complexity 4](#_Toc71897991)

[How Selection Sort works with original diagram 4](#_Toc71897992)

[Insertion Sort 4](#_Toc71897993)

[Introduction 4](#_Toc71897994)

[Space and time complexity 4](#_Toc71897995)

[How Insertion Sort works with original diagram 4](#_Toc71897996)

[Implementation and Benchmarking 4](#_Toc71897997)

[Results of the benchmarking 4](#_Toc71897998)

[Summary of Results- Graph and Table 4](#_Toc71897999)

# Preliminaries

For this project you will write an application which will be used to benchmark five different sorting algorithms. You will also write a report which introduces the algorithms you have chosen, and discusses the results of the benchmarking process. The five sorting algorithms must be chosen according to the following criteria:

1. A simple comparison-based sort (Bubble Sort, Selection Sort or Insertion Sort)

2. An efficient comparison-based sort (Merge Sort, Quicksort or Heap Sort)

3. A non-comparison sort (Counting Sort, Bucket Sort or Radix Sort)

4. Any other sorting algorithm of your choice

5. Any other sorting algorithm of your choice.

For the purpose of this assignment brief the following Algorithms were chosen:

* Bubble Sort
* Merge Sort
* Counting Sort
* Selection
* Insertion

# 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.

Time efficiency considers the time or number of operations required for the computer takes to run a program (or algorithm in our case) • Space efficiency considers the amount of memory or storage the computer needs to run a program/algorithm

# Sorting Algorithms

## Bubble Sort

### Introduction

### Space and time complexity

### How Bubble Sort works with original diagram

## Merge Sort

### Introduction

### Space and time complexity

### How Merge Sort works with original diagram

## Counting Sort

### Introduction

### Space and time complexity

### How Counting Sort works with original diagram

## Selection Sort

### Introduction

### Space and time complexity

### How Selection Sort works with original diagram

## Insertion Sort

### Introduction

### Space and time complexity

### How Insertion Sort works with original diagram

# Implementation and Benchmarking

Description of the process when implementing the application

## Results of the 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

## Summary of Results- Graph and Table