**Code Design and Data Structures**

Assessable Exercise: Searching and Sorting

Nicole Dance (Brown)

Contents

[Task 1 – Sorting 3](#_Toc17119841)

[Bubble Sort Algorithm 3](#_Toc17119842)

[Why? 3](#_Toc17119843)

[Advantages 3](#_Toc17119844)

[Disadvantages 3](#_Toc17119845)

[Algorithm 3](#_Toc17119846)

[Why? 3](#_Toc17119847)

[Advantages 3](#_Toc17119848)

[Disadvantages 3](#_Toc17119849)

[Algorithm 3](#_Toc17119850)

[Why? 3](#_Toc17119851)

[Advantages 3](#_Toc17119852)

[Disadvantages 3](#_Toc17119853)

[Task 2 Searching 3](#_Toc17119854)

# Task 1 – Sorting

## Bubble Sort Algorithm

### Why?

Bubble sort was selected as one of my 3 sorting algorithms as it is one, we have covered. Not only that but also due to the small amount of data in use for this assessment, Performance was not a major issue to consider. If I was to create an array of many more elements another algorithm would have been preferred. One with increased performance.

### Advantages

* Popular a simple to implement
* Works well with Arrays

### Disadvantages

* Slow due to comparing all elements against all other elements.
* Not very efficient

## Insertion Sort Algorithm

### Why?

Also, another algorithm covered. Again, used as there isn’t a high number of elements in the array. Also, whilst the same speed as bubble sort in its worst case. Otherwise, it is slightly faster. As it does not have to go through all the previous sorted elements. But instead goes through them until it finds the correct spot to insert it.

### Advantages

* Quicker Then bubble sort in its best case due to swapping fewer elements
* Simple to implement

### Disadvantages

* Slow, and as the array grows it will take longer
* Does not perform as well as other algorithms

## Selection Sort Algorithm

### Why?

Works by finding the minimum value in the array and then swapping it with the current position. I choose this algorithm as it is another one that works well with dynamic arrays.

### Advantages

* Works well on small lists
* Simple to implement

### Disadvantages

* Not very efficient for large lists
* Quicksort would have been a more efficient sorting choice.

# Task 2 Searching

### Why?

For this, I implemented the Binary Search Algorithm. In my program when the use selects search, I first run one of the Sorted Algorithm as Binary Search requires a sorted array. Binary search works well for a dynamic array as you can easily set the upper lower and middle bounds of the search and move accordingly.

### Advantages

* Unlike linear search which compares all elements in the array until the value is found. Binary search constantly halves the number of elements searched until the value is found.
* Binary Search is a simple algorithm to implement

### Disadvantages

* Requires a contain that supports random access. Which is required to grab the lower upper and middle values. It would not work so well for a linked list.
* Data must be sorted for a Binary Search to work. For unsorted Data linear search works but it slower.