­­­­­­

**SCHOOL OF SCIENCES & HEALTH**

***Department of Computing***

**Bachelor of Science in Computing**

**Programme Code: (DT211) 2013-14**

**YEAR 1**

**Name of module: Introduction to Algorithms**

**Name of Lecturer: Mr Martin McHugh**

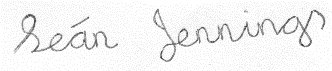
|  |  |
| --- | --- |
| **Student Names** | Seán Jennings |
| **Assignment title** | Bubble Sort and Improved Bubble Sort Algorithms |
| **Date issued** | 24th March 2014 |
| **Date Due for return** | 7th April 2014 |
| **Date Returned** | 7th April 2014 |

# Declaration

*(Optional content complete if requested by supervisor)*

I hereby certify that the material, which is submitted in this assignment/project, is entirely my own work and has not been submitted for any academic assessment other than as part fulfilment of the assessment procedures for the programme Bachelor of Science in Computing (DT211).

Signatures of student:



………………………...............................................................................................................................

Date: 24th March 2014

Table of Contents

[Declaration ii](#_Toc384558735)

[Table of Figures iv](#_Toc384558736)

[**1.** Comments on Code 1](#_Toc384558737)

[**2.** Tables 1](#_Toc384558738)

[**3.** Graphs 2](#_Toc384558739)

[**4.** Comments on Results 3](#_Toc384558740)

[4.1 Bubble Sort 3](#_Toc384558741)

[4.2 Improved Bubble Sort 3](#_Toc384558742)

# Table of Figures

[Figure 1: Bubble Sort Table 1](#_Toc384558890)

[Figure 2: Improved Bubble Sort Table 1](#_Toc384558891)

[Figure 3: Bubble Sort - Number of Comparisons vs. Number of Elements 2](#_Toc384558892)

[Figure 4: Bubble Sort - Number of Swaps vs. Number of Elements 2](#_Toc384558893)

[Figure 5: Improved Bubble Sort - Number of Comparisons & Number of Swaps vs. Number of Elements 3](#_Toc384558894)

# Comments on Code

* Both the regular and improved bubble sort algorithms use five arrays with 800, 1600, 2400, 3200 and 4000 elements respectively.
* The arrays in the regular bubble sort algorithm are initialised with n elements in descending order from n to 1.
* The arrays in the improved bubble sort algorithm are initialised with n elements that are random numbers in mod 2\*n.
* Both the regular and improved bubble sort algorithms send each n case to their respective sorting functions. Both sorting algorithms use counters for the amount of comparisons and swaps to display how the improved bubble sort is more efficient than the regular bubble sort.

# Tables

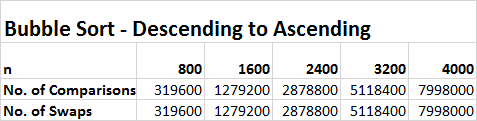


Figure : Bubble Sort Table

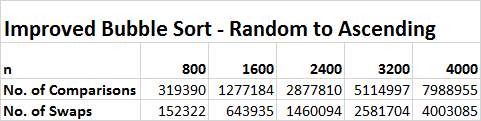


Figure : Improved Bubble Sort Table

# Graphs

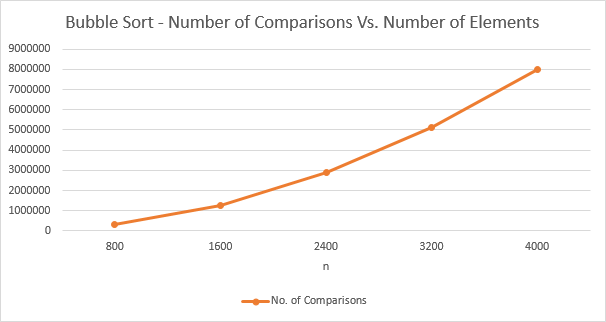


Figure : Bubble Sort - Number of Comparisons vs. Number of Elements

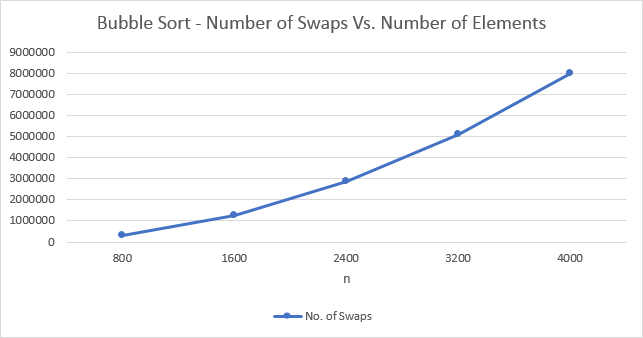


Figure : Bubble Sort - Number of Swaps vs. Number of Elements

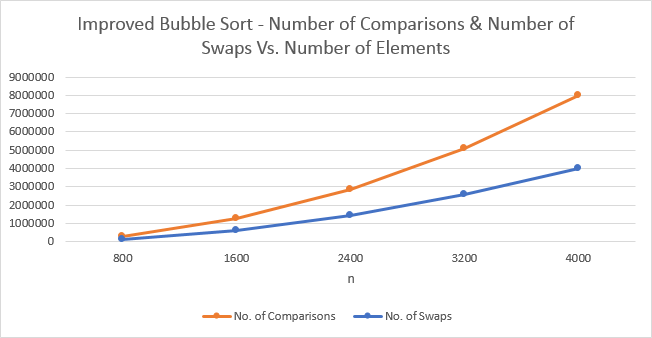


Figure : Improved Bubble Sort - Number of Comparisons & Number of Swaps vs. Number of Elements

# Comments on Results

## Bubble Sort

* The regular bubble sort algorithm compares each number in the array with the number next to it, and swapping them if needed.
* The algorithm does this repeatedly until it passes all the way through the array without swapping any numbers i.e. when they are in ascending order.

## Improved Bubble Sort

* It is not always necessary to make N-1 passes through the array as parts of each array can be sorted already i.e. there is a chance that some groups of the random numbers are already in ascending order.
* By using a Boolean variable the program knows that if no swap occurs, then array must be sorted, and it can break out of the outer for loop. This means that there are significantly less than N-1 passes.
* The graphs (figures 3 and 4) clearly show this as the number of swaps resulting from the improved algorithm is consistently halved when the compared to the regular algorithm.