

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

## **Computer Science Final Report**

*“Algorithmic Comparison Of Bubble Sort, Merge Sort And Heap Sort Using Software Benchmarking Tools”*

**Ryan Murray - C00180867**

**Derek Reilly - C00128213**

## **Introduction**

The purpose of this document is to demonstrate the product that was built for this project. This document will discuss the findings of the project and come to a conclusion on the findings. The student will discuss what was learned from completing project and what could have been done differently

# **Project Description**

## Purpose

The goal of this project was to use software benchmarking tools to analyse a number of algorithms and rate them on their performance. The algorithms would be tested for a number of attributes including reliability, robustness, security, cost, time, and number of elements.

## Components And Algorithms

The end product of this project is an executable which is a java program used to generate sets of elements of integers and sort the elements using numerous different sorting algorithms. The program can compare the performance of these algorithms against the standard java array sorting algorithm.

### Graphical User Interface

The project has a GUI interface which ties together all the other components of the project. The GUI was built using Java AWT which is Java's abstract window toolkit(*AWT Tutorial*, 2017). The AWT is collection of libraries for building platform independant graphical user interfaces.

The GUI is a very simple interface which only contains some checkbox's, buttons and text areas for displaying results.

### Quick Sort Algorithm Class

The quick sort algorithm is within a java class which is just for executing the algorithm on a supplied array of elements. This algorithm was extracted from Vogella tutorial on how to implement a quicksort algorithm in java (GmbH, 2010)

### Heap Sort Algorithm Class

The heapsort algorithm is within a java class which is just for executing the algorithm on a supplied array of elements. This algorithm was extracted from a tutorial on implementing an array based heap sort in java from the code 2 learn website (khwaja, 2011)

### Bubble Sort Algorithm Class

The Bubble sort algorithm is within a java class which is just for executing the algorithm on a supplied array of elements. This algorithm was extracted from a tutorial on implementing an array based bubblesort in java from the java T point website (Javatpoint, 2011)

## System Sort Algorithm Class

The system sort algorithm class is a java class which implements the standard java array sort algorithm. Java uses a 2 pivot quicksort implementation to perform sorts on arrays

## Algorithm Runner Class's

There are four java class's within the project which are designed to run the element generators and pass the elements into the sort algorithms to be sorted. These classes also implement the system algorithm. The classes also record the time taken to process the data and record the system resources used while performing the sorts.

## Elements generators

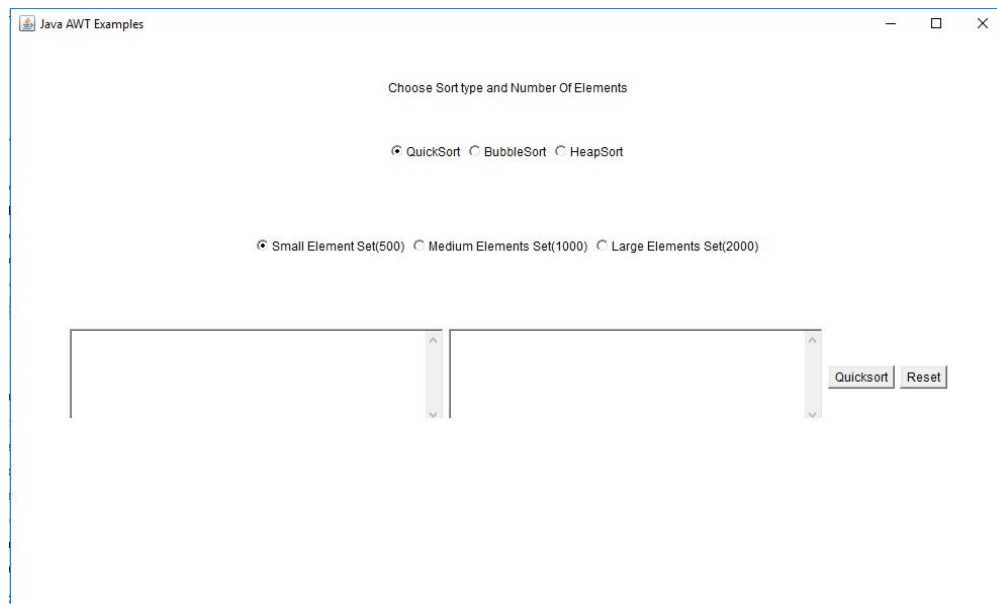
The elements generators are contained within the algorithm runner classes. These methods will generate a supplied number of random integers and store these in an array for sorting

## Java AWT Class

The AWT class is the gui component of the project. This class generates a graphical user interface allowing the user to select options for algorithms and element numbers they would like to process. This class calls the element generator and algorithm runner class's to perform the sorts based on the choices the user makes. Text areas are used to display the results of the sorts and the comparisons with the system sorts on the same set of elements.

## GUI

The GUI is very basic in design but includes everything necessary to perform what it sets out to do.



## **Learning Outcomes**

Through Completing this project a number of things were learned by the student about the implementation and differentiation of algorithms and also using Java.

### Java

Although the student used java before and completed modules for learning java the project was like a refresher course on using java. Java is a excellent language which can be picked up again very easily even after a long break from using it.

### Algorithms

By carrying out the implementation of the algorithms a better understanding was gotten of how the sorts actually function and the different components within the algorithms.

### Java AWT

As the student never used AWT before this was a good learning experience on the different components of the abstract window toolkit and how to construct a gui using the components. Although the AWT is now outdated and has been replaced with Swing and Fx a lot of components of these new libraries still use AWT components.

## **Success Analysis**

The purpose of the project was to build an application which could be used ot analyse different algorithms and compare the performance of these lgorithms to built in functions for sorting elements. Based on this goal the project could be considered a success.

Some data was collected using the application on the performance of the algorithms which can be used to draw a conclusion on the built in sort functions.

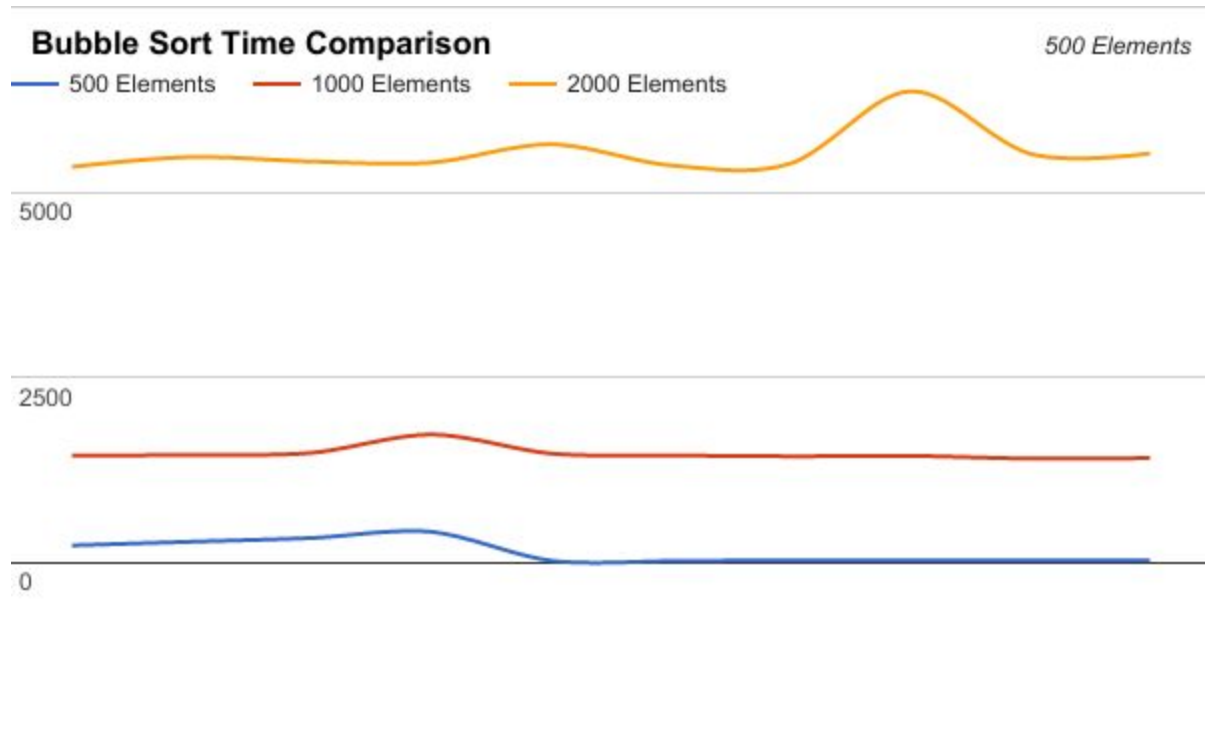
The anaysis were based on three different data sets , these were arrays of randomly genrated integers in the range of 1-10. There were three differents amounts of elements used to perform the tests, these were 500, 1000 and 2000. There were noticable differences in the performance of these algorithms

## Heap Sort vs Built In Sort



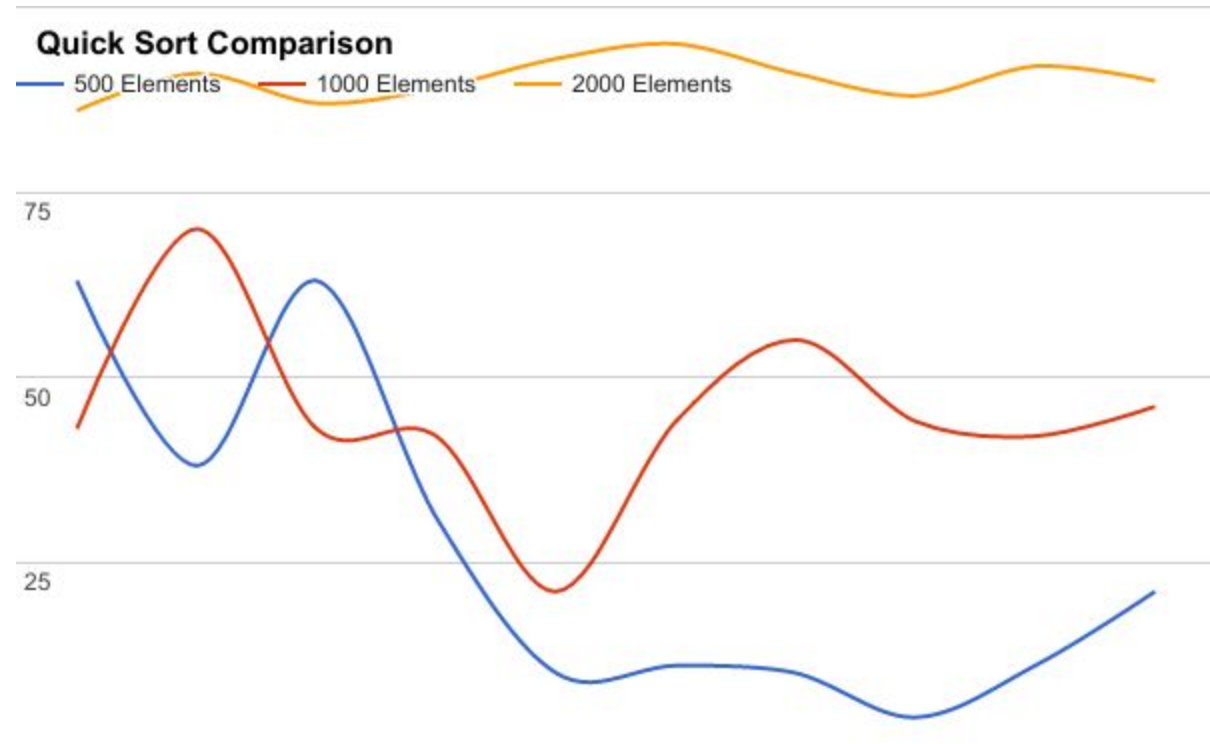
The chart above maps the difference in the time taken to perform the implemented heap sort vs the in built array sort function. It can be seen that as the number of elements increased the difference in time grew larger, and this difference was fairly consistent as the test was repeated. There was no difference recorded in the memory used by the 2 algorithms

## Bubble Sort Vs Built In Sort



It can be seen from the above graph that there is a large difference in time when the elements set gets larger between the bubble sort and the built in sort.

## Quick Sort Vs Built in Sort



The quick sort showed the least overall difference in time compared to the built in sort. This may be due to the fact that the built in sort is based on a quick sort algorithm which uses two pivots.

## Conclusion:



## **References**

AWT Tutorial (2017) Available at: <http://www.tutorialspoint.com/awt/> (Accessed: 3 March 2017).

GmbH, vogella (2010) *Quicksort in java - Tutorial*. Available at: <http://www.vogella.com/tutorials/JavaAlgorithmsQuicksort/article.html> (Accessed: 3 March 2017).

khwaja, farhan (2011) *HeapSort ( array based) implementation in java*. Available at: <http://www.code2learn.com/2011/09/heapsort-array-based-implementation-in.html> (Accessed: 3 March 2017).

Javatpoint (2011) *Bubble sort in java*. Available at: <http://www.javatpoint.com/bubble-sort-in-java> (Accessed: 3 March 2017).