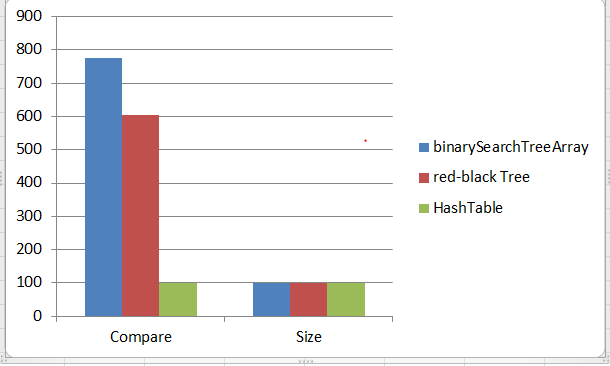
|  |
| --- |
| COMP09044 |
| Algorithms and Collections Project Part B |
| Individual Report |
|  |
| **Paul McLaughlin – B00268411** |
|  |

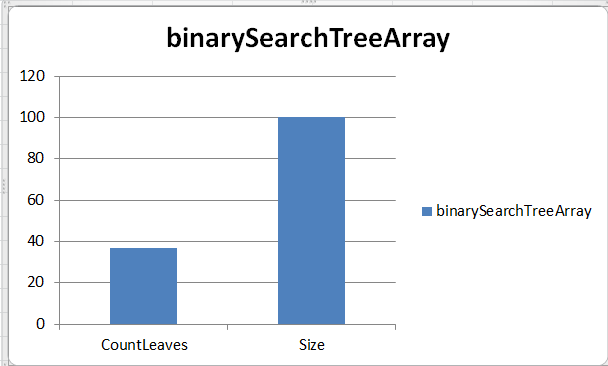
|  |
| --- |
|  |

# Introduction

This report is being made to present the results of tests carried out when running the project, it will discuss expected results with focus on the binary tree theorem, and also what the results actually were, and will have tests to show using both even and odd numbers, to compare what happened when switching between the two. Also, the report will go on to discuss why recursion was used for methods like countLeaves(), talking about how these recursive methods could have been implemented iteratively, explaining why, for this project, the final version was implemented with those recursive methods as opposed to iteratively, a critical appraisal and finally, a conclusion, summarising everything discussed in the report.

# Experiments





Shown above are results of the tests conducted for this project. The number of leaves Is 37 and the height is 14 for the binary search tree and for red-black tree the height is 7. The binary search tree theorem states a relationship between height and number of leaves: leaves(t) <= (n(t) + 1)/2.0 <= 2height(t), and this shows that these test results are supported within the binary tree theorem.

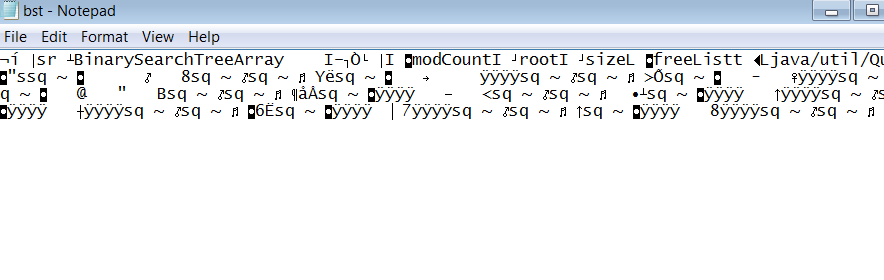
# Recursive vs Iterative

The reason for using recursion on this project for methods such as countLeaves(), is because it makes it easier to design algorithms and prove them correct, Due to the fact that recursion is usually less code than doing things in an iterative way, it’s also, usually less error prone. Thinking recursively breaks a problem up into a smaller version of that problem.

The recursive methods in the project could be made iterative by first, removing any recursive calls to methods, making a loop around the function body, for example, “While (true);”, since there is no tail recursion optimisation in java, It is best to leave it for performance purposes.

# Critical Appraisal

Overall, I am satisfied with what was managed in the project, I was able to write an application to test a lot of the different features of the binarySearchTreeArray class, it tested “add”, “height”, “compare”, “display”, “count”, “clear”, and “size”. The disappointment from this project was not being able to get the serialization part working correctly, I was able to create a file and the program did write something to it, but it looked like this:



The problem came from exceptions, and “Naïve methods” that I was unsure on how to fix, but, apart from that, the code runs with no errors and I am happy with the project.

# Conclusion

# 

In this report the experiments were delivered as graphs plotting different statistics from the project test results, these results were then discussed and noted to be supported by the binary tree theorem. Then went on to discuss recursion and why, in this project, it was used instead of iteration, mentioning ways in which methods of this project could have been implemented iteratively. Finally, a critical appraisal highlighting the part of the project that were complete, and the parts that were sadly incomplete, with a mention about my satisfaction with the project overall.

# 