



Thesis Fundamentals

**Student Name:** Kevin Colohan

**NUI Galway ID Number:** 15232065

**Course:** MCT624 Thesis Fundamentals

**Workshop No:** Week 6

**Assignments:** Assignment 1 – Thesis Proposal

**Date of Submission:** 27th/09/2016

Contents

Table of Contents

[1.0 Assignment 1: Thesis Proposal. 2](#_Toc462165593)

[1.1 Cover page 2](#_Toc462165594)

[2.0 Thesis Title 2](#_Toc462165595)

[3.0 Thesis Statement 3](#_Toc462165596)

[4.0 Scope and Significance of Research 4](#_Toc462165597)

[5.0 Methodology Selection 5](#_Toc462165598)

[6.0 Success Criteria for Thesis completion 6](#_Toc462165599)

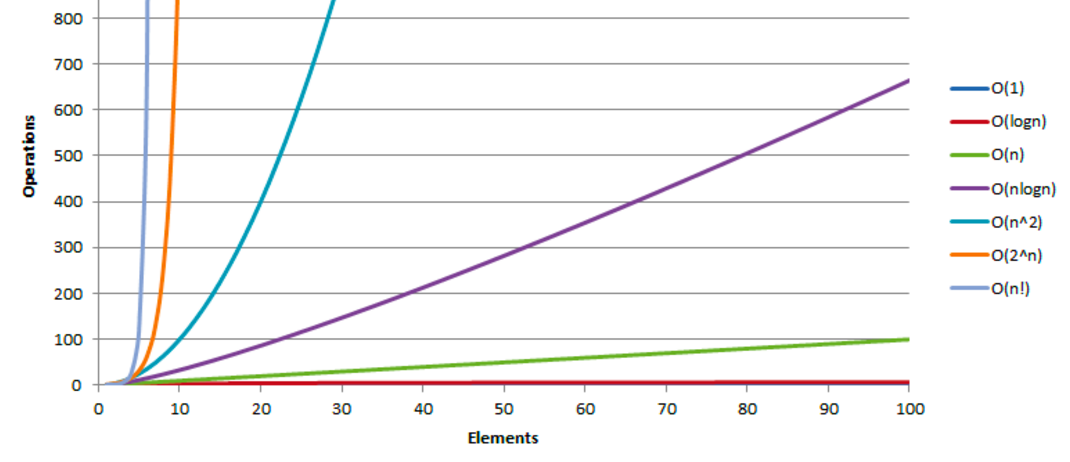
[7.0 Project Plan (week 7) 7](#_Toc462165600)

[8.0 References 7](#_Toc462165601)

Appendix 1 – Assignment 1 Source code

# Assignment 1: Thesis Proposal.

## Cover page



## Thesis Title

A study into search & sort algorithm complexity to understand how complexity effects performance runtime.

## Thesis Statement

In the field of computer science the big-O notation is used to describe the theoretical performance of an algorithm. Big-O measurement is usually made to measure the time or memory consumption used by an algorithm.

In this thesis by explaining and documenting the big-O performance of a number of selected constant O(1), logarithmic O(log n), sub-linear O(nd), linearithmic O(n log n), quadratic (n2) and exponential O(2n) algorithms will help future software developers answer the old question of, “Why is my program going so slow?”

By writing a logarithmic O(log n), linear log N and exponential O(2n) search & sort algorithms in C language, will allow for the analysis of the algorithms.

Using the scientific method, observe, hypothesize, predict, verify and validate to understand their complexity. By understanding and comparing the complexity of the algorithms I will answer the basic question of “How complexity effects performance”.

These results will be documented and in the future this research may aid future developers when implementing the optimal algorithm for their applications.

## Scope and Significance of Research

First question to answer is, what is an algorithm? Cormen and Leiserson describe it as:

“Informally, an algorithm is any well-defined computational procedure that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm is thus a sequence of computational steps that transform the input into the output.” **(Cormen; Leiserson; 2009)**

Algorithms are in widespread use within commercial applications. Some examples include video compression algorithms that are used in DVD, Blu-ray, YouTube and other video digital systems. Other uses of algorithms are encryption algorithms for security purposes to seismic modeling used within the resource exploration, traffic models, financial models etc.

The algorithm itself should have a well-defined set of instructions to solve the problem. It should be effective meaning that it should solve the given problem that it was originally designed for. A hopeless algorithm is one that never ends trying to solve a problem. Running time is the key, the idea of running time is how long a machine takes to achieve a computational result. This concept is mentioned by a man called Charles Babbage in 1864,

*"As soon as an analytic engine exists, it will necessary guide the future course of science. Whenever any result is sought by its aid, the question will arise, By what course of calculation can these results be arrived at by the machine in the shortest time?"* (Morrision 1989)

Complexity is a tool that allows us explain how an algorithm behaves when the input grows. If a larger input is given, how does a given algorithm behave? E.G. if it only take 1 second to run from an input of a size of 10,000 how will the same algorithm behave when the input size is doubled or tripled. Does the algorithm have the same run time or does it have a slower or faster run time behavior? Within programming the algorithms performance is import to know in advance before selecting what algorithm to use for an application job. The analysis of the complexity of a given algorithm will give data on how long the applications code will run for the largest cases. By analyzing and measuring the behavior for small input we can evaluate what will happen for larger inputs.

E.G. a binary search is a fast algorithm when searching sorted arrays.

The below example enables us to find an index of an element of a correct block in O (lg n) time. By understanding how this algorithm operates will allow us to understand the best use for such a search and if any improvements can be made to it. These results might then be used within the IT industry for further improvements on existing development projects. By fully understanding & documenting these performance behaviors will allow future developers make the correct choice with implementing such algorithms.

int binary\_search(item\_types s[], item\_type key, int low, int high){

int middle;

if (low>high) return(-1);

middle = (low+high)/2;

if (s[middle] == key(middle);

if ( s[middle] > key)

return(binary\_search(s,key,low,middle-1));

else

return(binary\_search(s,key,middle+1,high))

}

(Skiena2012)

## Methodology Selection

The approach with this thesis will be of a quantitative research. What I am trying to achieve over the past few weeks is to write a logarithmic O(log n), linear log N and exponential O(2n) search & sort algorithms in C language. Once these are completed by mid-October a sweet of analyzing tools will evaluate the performance on each of the selected algorithms. Data will be collected and evaluated. Then and only then can the question of how complexity effects runtime can be properly examined.

Observations: = running time of an application on a computer.

e.g. If N = 100, the approximate completion runtime for the following distinct categories of algorithms shows,

|  |  |  |
| --- | --- | --- |
|  | O(Log(N)) | 10-7 seconds |
|  | O(N) | 10-6 seconds |
|  | O(N\*Log(N)) | 10-5 seconds |
|  | O(N2) | 10-4 seconds |
|  | O(N6) | 3 minutes |
|  | O(2N) | 1014 years. |
|  | O(N!) | 10142 years. |

Hypothesize: = a model that is consistent with the observations.

Predict: = events using the hypothesis.

Verify: = the predictions by making further observations.

Validate: = by repeating until the hypothesis and observations agree.

Experiments must be reproducible.

Hypotheses must be falsifiable.

## Success Criteria for Thesis completion

This thesis will be completed when I fully document the results of how complexity determines performance.

## Project Plan (week 7)

## References

**Thomas H. Cormen, Chales E. Leiserson (2009),**[***Introduction to Algorithms 3rd edition***](http://www.amazon.com/dp/0262033844)***.***

Philip Morrison; Emily Morrison; Charles Babbage(1989),Charles Babbage On the principles and development of the calculator, Retrieved from: [https://books.google.co.uk/books?id=RUDdYIL5TG0C&pg=PA69&lpg=PA69&dq=%22As+soon+as+an+analytic+engine+exists,+it+will+necessary+guide+the+future+course+of+science.+Whenever+any+result+is+sought+by+its+aid,+the+question+will+arise,+By+what+course+of+calculation+can+these+results+be+arrived+at+by+the+machine+in+the+shortest+time?&source=bl&ots=VwrJvnIEj5&sig=Z2o9PEVL0CVStEUJ8V4SqaBKOxg&hl=en&sa=X&ved=0ahUKEwiYqoXgyebOAhXCL8AKHewiBw8Q6AEILzAD#v=onepage&q=%22As%20soon%20as%20an%20analytic%20engine%20exists%2C%20it%20will%20necessary%20guide%20the%20future%20course%20of%20science.%20Whenever%20any%20result%20is%20sought%20by%20its%20aid%2C%20the%20question%20will%20arise%2C%20By%20what%20course%20of%20calculation%20can%20these%20results%20be%20arrived%20at%20by%20the%20machine%20in%20the%20shortest%20time%3F&f=false](https://books.google.co.uk/books?id=RUDdYIL5TG0C&pg=PA69&lpg=PA69&dq=%22As+soon+as+an+analytic+engine+exists,+it+will+necessary+guide+the+future+course+of+science.+Whenever+any+result+is+sought+by+its+aid,+the+question+will+arise,+By+what+course+of+calculation+can+these+results+be+arrived+at+by+the+machine+in+the+shortest+time?&source=bl&ots=VwrJvnIEj5&sig=Z2o9PEVL0CVStEUJ8V4SqaBKOxg&hl=en&sa=X&ved=0ahUKEwiYqoXgyebOAhXCL8AKHewiBw8Q6AEILzAD#v=onepage&q=%22As%20soon%20as%20an%20analytic%20engine%20exists%2C%20it%20will%20necessary%20guide%20the%20future%20course%20of%20science.%20Whenever%20any%20result%20is%20sought%20by%20its%20aid%2C%20the%20question%20will%20arise%2C%20By%20what%20course%)

Steven S. Skenia (2012), The Algorithm Design Manual 2nd Edition; 4.9 Binary Search and Related Algorithms pages 132- 133

**Appendix 1**

**Assignment 1**

**Source Code**