**Time Complexity**

**Document Id 6**

**Top Level Learning Goals**

1 Understand Time Complexity O(n) , Θ(n, m)

2 Understand how to apply O(n)

**Section 1 Big O notation**

Learning Objectives

* **Big O**  high level forms O(1) , O(n), O(n^2)
* **Big O**  patterns
* **Big O**  specific calculation

**Section 1 Guided Learning Tasks**

* Limits
* Patterns
* Working with Sequence Approximations

Lab 1 Practical Obtain Sequence Solutions

Lab 1 Solution

**Section 2**

**Working with Big O**

Learning Objectives

* Array Operations
* Time complexity of an implicit Data Structure operation
* Sorting Array complexity

Section 2 Guided Learning Tasks

* Array Operations 1 Equilibrium Index
* Array Operations 2 Optimal Index
* Sorting Arrays

Lab 2 Review Questions

Lab 2 Practical Array task with Expected Worst case time complexity O(n)

Lab 2 Solution

**Section 3**

**Working with O(n) Advanced**

Learning Objectives

* Space complexity
* Big Theta (Θ)
* Sorting Algorithm

Section 3 Guided Learning Tasks

* Sorting Algorithms Time and Space complexity
* Common Sorting Algorithms best and worst case

Lab 3 Review Questions

Lab 3 Practical Sorting Algorithm efficiency Solution

Lab 3 Solution

Further Reading

http://ssp.impulsetrain.com/big-o.html

<https://philipstel.wordpress.com/2011/03/07/determining-the-complexity-of-an-algorithm-the-basic-part/>

<http://bigocheatsheet.com/>

https://www.reddit.com/r/dailyprogrammer