

Short Title:	Data Structs & Algos for NLP APPROVED
Full Title:	Data Structures & Algorithms for Natural Language Processing
Module Code:	MHLT H6014
ECTS credits:	10
NFQ Level:	9
Module Delivered in	1 programme(s)
Module Contributor:	Simon McLoughlin
Module Description:	The aim of this module is: To provide students with in-depth skills and knowledge of the role of advanced data structures in software engineering, in particular their design, analysis and correctness. To give students the skills necessary a wide range of algorithmic techniques for software development. To provide students with the necessary theoretical and practical applications framework for software engineering for natural language applications. To instil an in-depth appreciation of the importance of quality in software development.
Learning Outcomes:	
<i>On successful completion of this module the learner will be able to</i>	
<ol style="list-style-type: none"> 1. Research, select and apply the appropriate data structures and algorithms to use in solving practical and complex problems in software design for natural language applications 2. Evaluate and differentiate between stacks, queues, trees and graphs and other data structures and apply the 'O notation' in evaluating algorithms performance and time / space complexity 3. Research and develop solutions associated with data structure choices in the provision of software solutions in natural language applications 4. Research and develop solutions associated with algorithm choices in the provision of software solutions in natural language applications 5. Explore and integrate the use of the graph and tree type data structures in text-based information retrieval 6. Research, design, implement and test Abstract Data Types including stacks, queues, linked lists, trees and graphs. 7. Implement and evaluate the performance of classical software algorithms including sorting algorithms, searching algorithms, tree and graph algorithms. 	

Module Content & Assessment

Indicative Content
<p>Analysis of Algorithms (10%) The place of data structures and algorithms in software engineering, The dimensions of computational thinking in algorithm design, Calculating time and space complexity, Calculating Worst case and average case, What does algorithm correctness mean?, The design and behaviours of classic and well known algorithms, The O notation / Time and Space complexity of programs, Complexity, inefficiency and intractability.</p>
<p>Data Abstraction and Abstract Data Types (10%) Specifying and Designing ADTs, The ADT List, The ADT Sorted List, An Array-based implementation of the ADT List.</p>
<p>Linked Lists and ADTs (10%) Resizable vectors and arrays, Reference-based linked lists and linked list operations, Passing a linked list to a method, Tail references, Circular Linked Lists / Doubly linked lists.</p>
<p>The Abstract Data Type Stack (10%) An array based implementation of the ADT Stack, A reference based implementation of the ADT Stack, Comparing implementations, Simple Applications of the ADT stack.</p>
<p>The Abstract Data Type Queue (10%) An array-based implementation, An implementation that uses the ADT List, A reference-based implementation, Comparing Implementations, Simple Applications of the ADT Queue.</p>
<p>The Abstract Data Type Binary Tree (10%) Operations of the ADT Binary Tree, Traversals of a Binary Tree, Possible Representations of a Binary Tree, A Reference-based Implementation of the ADT Binary Tree, The ADT Binary Search Tree and BST operations.</p>
<p>The Abstract Data Type Graph and Graph Algorithms (20%) Graph Representations - Adjacency Matrix and Adjacency List, Graph Traversals, Minimum Spanning Tree Algorithms, Shortest Path Algorithms, Directed Graphs.</p>
<p>Sorting and Searching Algorithms (10%) Sequential search and binary search, Insertion Sort / Merge Sort, Analysis of Sorting and Searching Algorithms.</p>
<p>Text Processing Algorithms (10%) String Matching, Text Processing, Tries, Finite State Automata.</p>

Indicative Assessment Breakdown	%
Course Work Assessment %	100.00%

Course Work Assessment %				
Assessment Type	Assessment Description	Outcome addressed	% of total	Assessment Date
Practical/Skills Evaluation	Practical exercises based on lecture material	1,2,3	20.00	Every Week
Project	Developing research skills	3,4	40.00	Week 6
Lab Workbook	Exercises in data structures	5,6,7	40.00	Week 10

No Final Exam Assessment %

Indicative Reassessment Requirement
<p>Coursework Only <i>This module is reassessed solely on the basis of re-submitted coursework. There is no repeat written examination.</i></p>

ITB reserves the right to alter the nature and timings of assessment

Indicative Module Workload & Resources

Indicative Workload: Full Time	
Frequency	Indicative Average Weekly Learner Workload
Every Week	24.00
Every Week	24.00
Every Week	152.00

Indicative Workload: Part Time	
Frequency	Indicative Average Weekly Learner Workload
Every Week	24.00
Every Week	24.00
Every Week	152.00

Resources

Recommended Book Resources

Kapetanios, Epaminondas., Doina Tatar and Christian Sacarea 2014, *Natural Language Processing: Semantic Aspects.*, CRC Press/Taylor & Francis Group Boca Raton, FL

Jeffrey J. McConnell 2008, *Analysis of algorithms*, Jones and Bartlett Publishers Sudbury, Mass. [ISBN: 9780763707828]

Frank M. Carrano, Janet J. Prichard 2006, *Data abstraction and problem solving with Java*, Pearson/Addison Wesley Boston [ISBN: 9780321304285]

Supplementary Book Resources

Daniel Jurafsky, James H. Martin, *Speech and language processing*, Upper Saddle River, NJ Pearson 2009 [ISBN: 0135041961]

Donald Knuth 1991, *Fundamental Algorithms*, Addison-Wesley

Jokers, Matthews 2014, *Text Analysis with R for Students of Literature (Quantitative Methods in the Humanities and Social Sciences)*, Springer

Ingersoll, Grant S., Thomas S. Morton and Andrew L. Farris 2013, *Taming Text: How to Find, Organize, and Manipulate It*, Manning Publications

Wataru Nakamura 2011, *New perspectives in Role and Reference Grammar.*, Chapter by Nolan and Salem, Cambridge Scholars Publishing Newcastle upon Tyne

This module does not have any article/paper resources

Other Resources

Paper: Brian Nolan 2004, *First Steps Towards a Computational Role and Reference Grammar*
http://linguistics.buffalo.edu/people/fa_culty/vanvalin/rrg/RRG2004%20Book%20of%20Proceedings.pdf

Paper: Nolan, Brian, Ricardo Mairal-Uson and Carlos Perriñán 2009, *Natural Language processing applications in an RRG Framework*, Proceedings of the 10th International Role and Reference Grammar Conference. University of California, Berkeley USA.

Module Delivered in

Programme Code	Programme	Semester	Delivery
BN_KMHLT_R	Master of Science in Computing in Multimodal Human Language Technology	2	Elective