جـــامـــعـــة مـحـمـد بـــن زايـــد
للـــذكـــاء الاصـــطـــنـــاعـــي
MOHAMED BIN ZAYED UNIVERSITY
OF ARTIFICIAL INTELLIGENCE

Core Courses Syllabi

CS702 - Data Structures and Algorithms

Title	Data Structures and Algorithms
Code	CS702
Loading	4 Credit-hours
Prerequisites	Programming skills in any language
Catalog Description	This course provides a comprehensive introduction to fundamental data structures and algorithms which are essential to design an efficient computer program. It includes basic and advanced concepts in data structures such as array-based list, linked lists, hash tables, binary trees. The course will also cover basic and advanced topics in algorithms. The students will learn the relationship between data structures, algorithms and programming, and will be introduced to various performance measures and analysis techniques.
Goal	This elective course aims to familiarize students with various concepts in data structures and algorithms which are essential to design an efficient computer program.
Content	The course covers the basic concepts, algorithms , implementation techniques and application of data structures and algorithms, with major focus on search trees, hash tables, heaps, Fibonacci heaps, union-find in data structure and string matching, sorting and ordering statistics, graph algorithms,network flows, dynamic programming and NP-completeness in algorithms .
Recommended Textbooks	1. Kenneth Lambert, Fundamentals of Python: Data Structures, 2nd edition, Cengage learning, 2019. 2. Pat Morin's. Open Data Structures. 2016. 3. Cormen, Leiserson and Rivest. Introduction to Algorithms. Third edition. The MIT Press.
Recommended References & Supplemental Material	Instructors will provide reading material for additional topics covered in the course.

Teaching Week	Topics
1	**Introduction to Data Structures and Algorithms** **Lecture** • Basic Data Structures and Algorithms • Review of essential mathematics • Correctness and complexity analysis **Lab** • Practice problems for correctness and complexity analysis
2	**Array and String** **Lecture** • Array Stack: static and dynamic allocation • Fast Array Stack, Array Dqueue, DualArrayDeque, RootishArrayStack • Reversing the order of words in a sentence, detecting a palindrome • Counting the number of words, number of repeated words within a string • String matching **Lab** • Instructor-led demonstration related to the topics taught in the week
3	**Linked Lists** **Lecture** • Singly Linked List: insertion, deletion, searching, traversing, reverse traversing • Doubly Linked List: insertion, deletion, reverse traversal • A Space-Efficient Linked List: space requirements, adding, finding and removing elements • Skiplists : basic structure and analysis **Lab** • Practice problem related to linked lists
4	**Binary Search Trees I** **Lecture** • Basic operation: searching, adding and removal • Recursive algorithms, traversing binary trees • Random binary search trees **Lab** • Instructor-led demonstration related to the topics taught in the week

Teaching Week	Topics
5	**Binary Search Trees II** **Lecture** • Scapegoat trees with partial rebuilding • Red-black trees: adding, removing, insertion and deletion of leaf • AVL trees: rotation, rebalancing, insertion and deletion **Lab** • Practice problem related to scapegoat trees, red-black trees, avl trees
6	**Heap, stacks and queues** **Lecture** • Binary heap and MeldableHeap • Ordered and unordered stacks • Standard, double and priority queues **Lab** • Instructor-led demonstration related to the topics taught in the week
7	**Hash tables** **Lecture** • ChainedHashTable • LinearHashTable • Hash Codes • Table doubling, Karp-Rabin • Hash Table Collision Resolution **Lab** • Instructor-led demonstration related to the topics taught in the week
8	**Sorting and Searching Algorithms** **Lecture** • Lower bounds for comparison-based sorting • Bubble sort,merge sort, quick sort, shell sort, heap sort, insertion sort (binsort, radix sort) • Sequential searching • Probability search **Lab** • Instructor-led demonstration related to the topics taught in the week

Teaching Week	Topics
9	**Divide-and-Conquer** **Lecture** • Fast integer multiplication, recurrences, fast matrix multiplication • Naive divide and conquer algorithm • Faster divide and conquer algorithm • Master theorem, randomized median and selection algorithms **Lab** • Instructor-led demonstration related to the topics taught in the week
10	**Graphs I** **Lecture** • Graph basics, representing graph, exploring graphs, connectivity • Graph Traversal: breadth-first search and depth-first search • Biconnectivity in undirected graphs • Connected components in directed graphs • Topological sorting **Lab** • Instructor-led demonstration related to the topics taught in the week
11	**Graphs II** **Lecture** • Fastest Route • Naive Algorithm • Dijkstra's Algorithm: • Bellman-Ford Algorithm • Minimum Spanning Trees: Kruskal's Algorithm, Prim's Algorithm **Lab** • Instructor-led demonstration related to the topics taught in the week
12	**Dynamic Programming** **Lecture** • Paradigm of SPs in DAGs • Longest increasing subsequence; (approximate) string matching • Integer and (0, 1) knapsack problems • Chain matrix multiplication • Single-pair reliable SPs, all-pairs SPs: independent sets • Fibonacci, shortest paths **Lab** • Instructor-led demonstration related to the topics taught in the week

Teaching Week	Topics
13	**Network Flows** **Lecture and Tutorial** • Maxflow-mincut theorem • bipartite matching • Menger's theorem and disjoint dipaths • Ford–Fulkerson Algorithm • Edmonds–Karp Algorithm • Global minimum cuts • Image Segmentation
14	**NP-complete Problems** **Lecture** • Brute force search • Approximation Algorithms and Fix Parameter Tractability • P and NP, Showing NP-completeness • Integer Linear Programming Problem • Traveling Salesman Problem **Lab** • Instructor-led demonstration related to the topics taught in the week
15	**Data structure for Integers** **Lecture** • Binary Trie: digital search tree • XFast Trie: searching in doubly-logarithmic iime • YFast Trie: doubly-logarithmic time SSet **Lab** • Instructor-led demonstration related to the topics taught in the week