الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

Data Structures & Algorithms II

Course Definition

Powered by: KNOWLGICA KNOWLEDGE SOLUTIONS & TECH

Syrian Arab Republic		الجمهورية العربية السورية
Ministry of Higher Education		وزارة التعليم العالـــي
Syrian Virtual University		الجامعة الافتراضية السورية

1. Basic Information:

Course Name	Data Structures & Algorithms II
Course Code	SDA601
Number of Presentational Sessions*	24-32
Number of Synchronous Sessions**	12-16
Number of Shorter Tests***	0
Number of Exams***	1
Theoretical Sessions Work Load (hrs.)	36-48
Practical Sessions Work Load (hrs.)	18-24
Credit Hours	5

***Each presentational session comprises both recorded lecture (1.5 hrs.) and interactive learning content (1.5 hrs.).**

****Each synchronous session comprises the interactive lecture carried out in real time in a virtual class (1.5 hrs.).**

*****Each shorter test is 0.5 hr. long. The final exam is 2 hrs. long.**

N.B.

Generally, each chapter requires two presentational sessions: one for the recorded content and one for the interactive content (unless the chapter is too long, in which case it may require more sessions (. This note applies to synchronous sessions as well, where each chapter requires one synchronous session generally.

2. Prerequisites courses:

Course	Code
Data Structures & Algorithms I	BDA501

3. Course Objectives:

The Data Structures and Algorithms (2) course, which is an extension of the Data Structures and Algorithms (1) course, deals with complements in binary trees, data (graphs), and some new sorting and searching algorithms. It also deals with non-binary tree data structures such as B-Trees, B + Trees, Red-Black Trees and operations on. In the Algorithms axis, it will be discussion about Greedy Algorithms and dynamic programming.

4. Learning Outcomes (LO):

By the end of this course, the learner is expected to acquire and learn the following subjects:

- Complete operations related to binary trees
  - Balanced Trees
  - Binary Search Trees
  - AVL Trees
- Complete operations related to Graphs
  - Topological arrangement
  - Spanning Trees
- Learning about Non-Binary Trees And Operations related To:
  - B-Trees
  - B+Trees
  - Red-Black Trees
- Learning about new Sort and Search Algorithms:
  - Counting Sort
  - Radix Sort
  - Bucket Sort
  - Heap Sort
  - …
- Learning about Greedy Algorithms:
  - Greedy Algorithm
  - Ford-Fulkesron Algorithm
  - Dijkstra's Algorithm
  - Kruskal's Algorithm
  - Prim's Algorithm

- Huffman Coding
- …..

- Learning about Dynamic Programming:
  - Dynamic Programming
  - Floyed Warshall Algorithm
  - Longest Common Subsequence
  - ……

5. Assessment Results:

Chapter Number	Chapter Title	General Objectives	Assessment Type				
			Interactive Content & Recorded Sessions	Applied Activities (Synch. Sessions)	Final Exam*/ Shorter Tests	Presentations and Interviews	Reports
CH1	Complements in binary trees	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH2	Complements in Graphs	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH3	Non-Binary Trees Structures (B-Trees)	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH4	Non-Binary Trees Structures (B+Trees)	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH5	Non-Binary Trees Structures (Red-Black Trees)	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH6	New Sort and Search Algorithms (Counting – Radix-Bucket and Heap Sort)	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH7	Greedy Algorithms (Greedy ,Ford-Fulkesron, Dijkstra Kruskal, and Prim, Huffman coding)	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√
CH8	Dynamic Programming, Floyed Warshall, Longest Common Subsequence	Comprehension -Analytical Thinking -Tools and Application Hands- On	√	√	√		√

***The final exam is two hours long and is given at the end of the course.**

****Shorter tests are about 30 minutes long and are given after three or four lectures throughout the semester during synchronous sessions.**

*****Presentations, interviews, and reports are submitted once after each three or four lectures throughout the semester during synchronous sessions.**

6. Course Syllabus:

Chapter	Subject	Content	Number of Learning Objects	Number of synchronous Learning Objects
CH1	Complements in binary trees	1. Introduction 2. Introducing balanced trees and explaining their characteristics and types. 3. Definition of binary search trees 4. Description of the basic operations on the binary search trees (search-create-add-delete). 5. Studying and analysis the Operations (search-create-add-delete) in the binary search tree. 6. Define AVL trees. 7. Description of the basic operations on AVL trees (sub-tree rotation - left and right rotation - add - delete). 8. 8. Study and analyze the algorithms of (sub-tree rotation-rotate right and left-add-delete) in the AVL tree. 9. Examples and exercises.	1	1
CH2	Complements in Graphs	1. Introduction 2. Topological arrangement 3. Spanning Trees and Basic Operations on.	1	1
CH3	Non-Binary Trees Structures (B-Trees)	1. Introduction 2. B-Tree trees 3. Definitions and Concepts in B-Trees	1	1

		4. Characterization of the basic operations on B-Trees (search-create-add-delete) 5. Studying and analysis (search-create-add-delete) algorithms in B-trees 6. Examples and exercises.		
CH4	Non-Binary Trees Structures (B+Trees)	1. B + Tree 2. Definitions and Concepts in B + Trees 3. A comparison between B-Tree and B + Tree 4. Characterize of the basic operations on B + Trees (find-create-add-delete) 5. Studying and analysis of (search - create - add - delete) algorithms in the B + Tree 6. 12. Examples and exercises.	**1**	**1**
CH5	Non-Binary Trees Structures (Red-Black Trees)	1. Introduction 2. Definition and characteristics of Red-Black trees 3. Operations on Red-Black trees (rotating sub-trees - rotating the red-black tree from right to left and left to right - adding an item to a Red-Black tree - removing an item from the Red-Black tree. 4. Examples and exercises. 5. Conclusion and Summary of Non-Binary Trees.	**1**	**1**
CH6	New Sort and Search Algorithms (Counting – Radix-Bucket and Heap Sort)	1. Introducing the Counting Sort algorithm, explaining the steps of the algorithm, and studying the time complexity. 2. Introducing the Radix Sort algorithm, explaining the steps of the algorithm, and studying the time complexity. 3. Introducing the Bucket Sort algorithm, explaining the steps of the algorithm, and studying the time complexity. 4. Introducing the Heap Sort algorithm, explaining the steps of the algorithm, and studying the time complexity. 5. Examples and exercises.	**2**	**1**
CH7	Greedy Algorithms (Greedy ,Ford-Fulkesron،	1. Introduction 2. Introducing Greedy Algorithm, explaining the steps of the algorithm, and studying	**2**	**2**

	Dijkstra Kruskal, and Prim, Huffman Coding Algorithm)	time complexity. 3. Introducing the Ford-Fulkesron algorithm, explaining the steps of the algorithm, and studying the time complexity. 4. Introducing Dijkstra's Algorithm, explaining the steps of the algorithm, and studying time complexity. 5. Introducing Kruskal's Algorithm, explaining the steps of the algorithm, and studying time complexity. 6. Introducing Prim's Algorithm, explaining the steps of the algorithm, and studying time complexity. 7. Introducing Huffman Coding's Algorithm, explaining the steps of the algorithm, and studying time complexity. 8. Examples and exercises.		
CH8	Dynamic Programming, Floyed Warshall, Longest Common Subsequence	1. Introduction 2. Define dynamic programming and explain how it works. 3. Comparison of dynamic programming with recursive algorithms and backtracking algorithms. 4. Definition of Floyed Warshall's algorithm and explain how it works. 5. Introducing Longest Common Subsequence (LCS) 6. Using dynamic programming to create LCS 7. Explanation the steps of the LCS algorithm. 8. Examples and exercises.	**1**	**1**

7. Practical Activity:

- **Tools and Labs:**

Tool Name	Description
C++	Programming Language

- **Practical Activities per Chapters:**

Chapter	Activities Type	Remarks
CH1	☑ Exercises ☑ Homework ☑ Webinars ☑ Project ☐ Experiment ☑ Other	
CH2	☑ Exercises ☑ Homework ☑ Webinars ☑ Project ☐ Experiment ☑ Other	
CH3	☑ Exercises ☑ Homework ☑ Webinars ☐ Project ☐ Experiment ☑ Other	
CH4	☑ Exercises	

	☑ Homework ☑ Webinars ☐ Project ☐ Experiment ☑ Other	
CH5	☑ Exercises ☑ Homework ☑ Webinars ☐ Project ☐ Experiment ☑ Other	
CH6	☑ Exercises ☑ Homework ☑ Webinars ☑ Project ☐ Experiment ☑ Other	
CH7	☑ Exercises ☑ Homework ☑ Webinars ☑ Project ☐ Experiment ☑ Other	
CH8	☑ Exercises ☑ Homework ☐ Webinars	

	☑ Project ☐ Experiment ☐ Other	

8. References:

- Introduction to Algorithm Second Edition. Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein .The MIT Press Cambridge, Massachusetts London, England