كلية الحاسبات و المعلومات

Analysis and Design of Algorithms Course Specifications

Faculty: Computer and Informatics.

Department: Information System

Program(s) on which the course is given : Bachelor in Computer and Information Sciences.

Major or Minor element of programs: All majors

Department offering the program : Information System

Department offering the course : Computer Science

Academic year / Level : 3rd Year / B.Sc.

Date of specification approval : 25/10/2009

A. Basic Information:

Title: Analysis and Design of Algorithms **Code:** CSC 340

Lectures: **3** hrs/week **Practical:** **2** hrs/week **Tutorial: ---**

Credit Hours: --- **Total:** **5** hrs/week

B. Professional Information:

1. Overall Aims of Course:

- Design algorithms using flowcharts.
- Define and analyze of the algorithms in varies styles, including recursive algorithms.

- Demonstrate a familiarity with major fundamental algorithms in sorting.
- Apply important algorithmic design paradigms and methods of analysis.
- Understand, analyze and apply standard algorithms involving tree and graph.
- Be able to design and evaluate algorithms in various styles, such as divide and conquer, greedy.
- Define and analyze the complexity of algorithms in various styles.

2. Intended Learning Outcomes of Course (ILOs):

a. Knowledge and Understanding:

a1- Know how algorithms are designed using flowcharts.

a2- Define and know what an asymptotic analysis of algorithm performance is.

a3- Describe and Understand all the methods used in algorithm complexity or performance: Big-O notation, powers, and logs, growth of functions, worst-, and best-case analysis of algorithms.

a4- Understand and recognize the fundamental algorithms that are used in sorting, and searching of data structures.

a5- Comprehend the fundamental types of algorithm design paradigms such as divide-conquer, greedy, and dynamic programming.

a6- Identify some useful applications of algorithms in the area of sorting, searching, and optimization problems.

a7- Know and understand abstract concepts about algorithm analysis and design, as these concepts are used in computing.

b. Intellectual Skills:

b1- Analyze and compare algorithms complexity using asymptotic analysis, and distinguish between the worst-, average-, and best-case analysis of algorithms.

b2- Conclude the major algorithms for sorting. Analyze and categorize the performance of these algorithms and the design strategies that the algorithms embody.

b3- Distinguish the different types of algorithm paradigms and evaluate when an algorithmic design situation calls for it.

b4- Define traditional and nontraditional problems, set goals towards solving them, and. observe results.

b5- Be able to test and debug simple computer programs.

b6- Generate an innovative design to solve a problem containing a range of commercial and industrial constraints.

b7- Restructuring solution methodologies up on their results.

b8- Program in the major computer programming paradigms.

c. Professional and Practical Skills:

c1- Design effective computer programs to solve a variety of scientific real-world problems.

c2- Apply greedy and dynamic-programming algorithms to solve practical scientific problems.

c3- Design new algorithms or modify existing ones for new applications and reason about the efficiency of the result.

c4- Deploy effectively the tools used for the construction and documentation of software, with particular emphasis on understanding the whole process involved in using computers to solve practical problems.

c5- Apply skills in developing an algorithmic solution to a problem and be able to represent it with appropriate program and data objects.

d. General and Transferable Skills:

d1- Present effective computer programs that employ efficient algorithms.

d2- Present a design of algorithms for solving scientific problems related to other disciplines.

d3- Present a project or a product.

d4- Work in stressful environment and within constraints.

d5- Manage tasks and resources.

d6- Apply improved problem solving skills to basic real world situations.

d7- Search for information and adopt life-long self-learning.

e. Attitude:

e1- A knowledge and respect of ethics and ethical standards in relation to a major area of study.

e2- Demonstrate an ethical behavior toward software copyrights

e3- Learn how to make relation with other, and the limit of this relation.

3. Contents:

Topic	No. of hours	Lecture	Tutorial/ Practical
Algorithms and flowcharts - I	**5**	**3**	**2**
Algorithms and flowcharts - II	**5**	**3**	**2**
Insertion sort, Analyzing algorithms and Designing algorithms	**5**	**3**	**2**
Asymptotic notation and Standard notations and common functions	**5**	**3**	**2**
Recurrences - I	**5**	**3**	**2**
Recurrences - II	**5**	**3**	**2**
Binary Search Trees - I	**5**	**3**	**2**
Binary Search Trees - II	**5**	**3**	**2**
Dynamic Programming - I	**5**	**3**	**2**
Dynamic Programming - II	**5**	**3**	**2**
Greedy Algorithms - I	**5**	**3**	**2**
Greedy Algorithms - II	**5**	**3**	**2**
Graph Algorithms - I	**5**	**3**	**2**
Graph Algorithms - II	**5**	**3**	**2**