University of Nicosia, Cyprus

Course Code	**Course Title**	**ECTS Credits**
COMP-370	Algorithms	6
Department	**Semester**	**Prerequisites**
Computer Science	Fall, Spring	COMP-211
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English
Level of Course	**Year of Study**	**Lecturer**
1st Cycle	4th	Dr Florent Domenach
Mode of Delivery	**Work Placement**	**Co-requisites**
Face-to-face	N/A	None

Objectives of the Course:

The main objectives of the course are to:

- Provide understanding how to evaluate the efficiency of an algorithm
- Present a variety of techniques for designing algorithms
- Provide a wide variety of data structures and should be able to use them appropriately to solve problems
- Build a foundation of fundamental algorithms

Learning Outcomes:

After completion of the course students are expected to be able to:

1. Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm.
2. Discuss factors other than computational efficiency that influence the choice of algorithms, such as programming time, maintainability, and the use of application-specific patterns in the input data.
3. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.
4. Determine the time and space complexity of simple algorithms.
5. Implement the most common quadratic and O(NlogN) sorting algorithms.
6. Design and implement an appropriate hashing function for an application.
7. Design and implement a collision-resolution algorithm for a hash table.
8. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.
9. Solve problems using the fundamental graph algorithms, including depth-first and breadth-first search, single-source and all-pairs shortest paths, transitive closure, topological sort, and spanning tree algorithm.
10. Demonstrate the following capabilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in programming context.

Course Contents:

1. Design and analysis of algorithms
    a. Running Time of a program
    b. Big Oh, big Omega, big Theta
2. Basic data types

a. List
b. Queues
c. Stacks

3. Trees
    a. Traversals
    b. ADT Tree
4. Basic operation on sets
    a. ADT Sets
    b. Hash Tables
    c. Priority Queues
5. Directed graphs
    a. Single Source Shortest Path
    b. All Pairs Shortest Path
    c. Transitive Closure
    d. Topological Sorting
6. Undirected graphs
    a. Traversals
    b. Minimum Spanning Tree
7. Sorting

Learning Activities and Teaching Methods:

Lectures, Practical Exercises and Assignments.

Assessment Methods:

Homework, Mid-Term, Final Exam.

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
S. Dasgupta, C. Papadimitriou and U. Vazirani	Algorithms	McGraw-Hill	2008	978-0-07-352340-8

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
R. Sedgewick	Algorithms	Addison-Wesley	1998	
R. Johnsonbaugh and M. Schaefer	Algorithms	Pearson	2004	
A. V. Aho, J. E. Hopcroft and J. D. Ullman	Data Structures and Algorithms	Addison-Wesley	1983	978-0201-0000238