GRAPH THEORY AND APPLICATIONS

COURSE OUTLINE

Responsible: Marios Antonakakis

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electronic Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0806.8.016.0	**SEMESTER**	1st
COURSE TITLE	Graph Theory and Applications		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	4	5
Total	**4**	**5**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Scientific Area, Skills Development
PREREQUISITE COURSES	Discrete mathematics
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English) — Spring Semester
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/EE136/

2. LEARNING OUTCOMES

Learning outcomes

The purpose of the course is to provide students with the theoretical knowledge and computational tools required to understand, analyse and apply Graph Theory to engineering problems and Information and Communication Technology applications.

Upon successful completion of the course, the student will be able to:

- Define, describe and represent graphs and their basic structural components (vertices, edges, weights, directionality, connectivity).
- Distinguish between different categories of graphs and select the appropriate representation (adjacency lists, adjacency matrices) according to the application.
- Design and implement basic graph traversal algorithms (e.g. BFS, DFS) and extract graph properties.
- Analyse the computational complexity of graph algorithms.
- Apply algorithmic techniques to solve problems in networks, information systems and telecommunications applications.
- Model real-world problems using graphs.

Evaluate and interpret the results of algorithmic analysis in order to select optimal solutions.

General Competences

Search, analysis and synthesis of data and information, using the necessary technologies

Adaptation to new situations

Decision-making

Autonomous work

Teamwork

Project planning and management

Exercise of criticism and self-criticism

Promotion of free, creative and inductive thinking

3. SYLLABUS

The aim of the course is to develop students' knowledge and skills in the theoretical understanding and computational implementation of graph algorithms using the Python programming language. The course focuses both on graph representation structures and on fundamental and advanced algorithmic techniques for processing graphs.

More specifically, the following topics are covered:

- Data structures for graph representation (adjacency lists and matrices).
- Graph traversals: depth-first search (DFS) and breadth-first search (BFS).
- Traversal of directed acyclic graphs (topological sorting).
- Graph connectivity and identification of connected components.
- Weighted graphs and distance metrics.
- Minimum Spanning Trees and the Prim and Kruskal algorithms.
- The Union-Find problem and the Ackermann function.
- Single-source and all-pairs shortest paths (Dijkstra, Bellman-Ford, etc.).
- Properties of paths and cycles.
- Trees and special classes of graphs.
- Connectivity, planarity and graph colouring.
- Perfect graphs and their fundamental properties.
- Applications of graph theory to networks, information systems, social networks and optimisation problems.

The course emphasises the implementation and analysis of algorithms using Python, as well as the evaluation of their computational complexity and practical performance.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Face-to-face theoretical teaching. Assignments in small student groups. Laboratory exercises on graph theory applications in small student groups. Laboratory training using Python.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Use of presentation software. Communication with students through an asynchronous e-learning platform. Use of Python and related packages for implementing graph algorithms and applications.

TEACHING METHODS The manner and methods of teaching are described in detail.	<table><tr><th>Activity</th><th>Semester workload</th></tr><tr><td>Lectures</td><td>39</td></tr><tr><td>Application Exercises in Small Groups</td><td>16</td></tr><tr><td>Laboratory Assigments</td><td>13</td></tr><tr><td>Invidual Application Exercise Assigments</td><td>26</td></tr><tr><td>Individual Study</td><td>52</td></tr><tr><td>Exams</td><td>4</td></tr><tr><td>Course total</td><td>150</td></tr></table>
STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	I. Written final examination (FE) (50%) II. Laboratory Examination (LE) (25%) III. Individual Application-Exercise Assignments (IAEA) (25%) The course grade (FE*0.5 + LE*0.25 + IAEA*0.25) must be at least 5. The grade for each of I, II and III must be at least five (5). The assessment criteria are available to students on the course website and are announced during the first class.

5. ATTACHED BIBLIOGRAPHY

- Recommended Bibliography:

Greek or translated textbooks:

- D. Dimitra, “Mathematics and Graph Theory for Engineers, 2nd Edition”, ISBN: 978-960-418-960-1, Eudoxus Code: 112690339, Year: 2022, Pages: 678
- S.D. Nikolopoulos, L. Georgiadis, L. Palios, “Algorithmic Graph Theory”, Greek Academic Electronic Textbooks and Aids, Hellenic Academic Libraries Link (HEAL-Link), 2015.
- Instructor’s notes.

Foreign-language textbooks:

- G. Ralucca, L. Craig, H. Stephen, “Graph Theory: Favorite Conjectures and Open Problems – 1”, Springer, 2016
- J. A. Bondy and U.S. R. Murty, “Graph Theory with Applications”, The Macmillan Press Ltd., 1982 (5th printing).
- R. Diestel, “Graph Theory”, Springer-Verlag, 2017 (5th edition).
- M. D. Penrose, “Random Geometric Graphs”, Oxford University Press, 2003.

- Related scientific journals:

Journal of Graph Theory, Wiley Online Library.

Selected articles from IEEE/ACM Transactions on Networking, IEEE Transactions on Biomedical Engineering, IEEE Transactions on Wireless Communications, etc.