COURSE DESCRIPTION

1. Program Information

1.1 University	**“**Alexandru **Ioan Cuza” University of Iaşi**
1.2 Faculty	Faculty of Computer Science
1.3 Department	Department of Computer Science
1.4 Study Domain	Computer Science
1.5 Study Cycle	**Undergraduate studies**
1.6 Study Program / Qualification	Computer Science/Computer Science Graduate

2. Course Information

2.1 Course Name			**Graph Algorithmics**				
2.2 Course Teacher			**Lect. Dr. Olariu Emanuel Florentin/**				
2.3 Seminary Teacher			**Lect. Dr. Olariu Emanuel Florentin, Lect. Dr. Diac Paul**				
2.4 Year of study	2	**2.5** Semester	2	**2.6** Evaluation	M	**2.**7 Course status*	OB

** OB – Compulsory / OP – Optional*

3. Total estimated hours (hours per semester and didactic activities)

3.1 Hours per week	**4**	in which: course **3.2**	**2**	**3.3** seminary	**2**
3.4 Hours in *curriculum*	**56**	in which: course **3.5**	**28**	**3.6** seminary	**28**

Time Distribution	hours
Manual study, Course support, Bibliography, and others	**20**
Supplementary Documentation in library, in electronic forums, and on the field	**10**
Seminaries/laboratories preparation, homeworks, reports, portfolios and essays	**35**
Tutoring	
Evaluation	**4**
Other activities (consultations per student)	

3.7 Total hours individual study	**65**
3.8 Total hours per semester	**125**
3.9 Credits	**5**

4. Preconditions (if necessary)

4.1 Of Curriculum	Data Structures, Algorithms Design
4.2 Of Skills	**--**

5. Conditions (if necessary)

5.1 For Course Operation	--
5.2 For Seminary Operation	

6. Specific Skills Acquired

Professional Skills	Upon successful completion of this discipline, students will be able to: **C1.** Understand and the use of specific concepts related to algorithmic graph theory. **C2.** Know the basic algorithms for solving the classic problems on graphs. **C3.** Have the capacity of designing and analysing new algorithms for problems on graphs.
Transversal Skills	Upon successful completion of this discipline, students will be able to: **C1.** Ddiscriminate between polynomial and nondeterministic polynomial classes of decision problems. **C2.** Have the capacity of estimating a time/space complexity for given algorithms. **C3.** Model real life problems as graph theory problems.

7. Course Objectives (from the grid of specific skills acquired)

7.1 General Objective	The taming of elementary and medium-advanced concepts and results from Algorithmic Graph Theory.
7.2 Specific Objectives	At the end of this course, the students will be able to: **O1.** Understand and explain basic notions like: vertices, edges, graphs, (spanning/ induced) subgraphs, trees, paths, cycles and connectivity, stable sets, matchings, flows and preflows, cuts, colorings, etc. **O2.** Describe and solve problems like: minimum cost paths, minimum cost spanning trees, maximum matchings, maximum flows/minumum cuts in a flow network, coloring the vertices of a graph etc. **O3.** Use and inter-connect the above concepts and problems. **O4.** Discriminate between NP and P problems; know a basic collection of NP-complete problems.

	Lectures	**Teaching Methods**	**Observations** (hours and bibliographic references)
1	Course description. Vocabulary of Graph Theory.	Exposition.	2, [1] - [5]
2	Vocabulary of Graph Theory.	Exposition.	2, [1] - [5]
3	Vocabulary of Graph Theory. Shortest path problems in (di)graphs.	Exposition.	2, [1] - [5]
4	Shortest path problems in (di)graphs.	Exposition.	2, [1] - [5]
5	Connectivity problems in (di)graphs (Menger's, Konig's, Hall's theorems). Spanning trees.	Exposition.	2, [1] - [5]
6	Minimum spanning trees problem (Prims's and Kruskal's algorithms). Maximum matchings and minimum edge-covers.	Exposition.	2, [1] - [5]

7	Maximum matching problem. Berge's and Tutte's theorems. Hopcroft Karp algorithm. Network flows.	Exposition.	2, [1] - [5]
8			
9	Network flows. Cuts, augmenting paths. Max flow - min cut theorem. Ford Fulkerson and Edmonds Karp algorithms.	Exposition.	2, [1] - [5]
10	Preflows (Ahuja Orlin algorithm). Combinatorial applications of flows in networks.	Exposition.	2, [1] - [5]
11	Minimum cost flows. Polynomial-time reductions for graph problems.	Exposition.	2, [1] - [5]
12	Polynomial-time reductions for graph problems. Approaching NP-hard problems.	Exposition.	2, [1] - [5]
13	Planar graphs. Drawing planar graphs.	Exposition.	2, [1] - [5]
14	Tree decomposition and applications.	Exposition.	2, [1] - [5]

Bibliography

Main references:

[1] Cormen T. H., C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd edition, MIT Press, 2009.
[2] Croitoru C., *Tehnici de baza in optimizarea combinatorie*, Editura Univ. "Al. I. Cuza", Iasi, 1992.
[3] Diestel R., *Graph Theory*, electronic edition.
[4] Lovasz L., *Combinatorial Problems and Exercises*, 2nd edition, North Holland, 1993.
[5] Tomescu I., *Probleme de combinatorica si teoria grafurilor*, Editura didactica si pedagogica, Bucuresti, 1981.

Supplementary references:

[6] Alon, N., J. H. Spencer, The probabilistic method, Wiley, 2008
[7] Croitoru C., *Introducere in proiectarea algoritmilor paraleli*, Editura Matrix Rom, Bucuresti, 2002.

	Seminary	**Teaching methods**	**Observations** (hours and bibliographic references)
1	Vocabulary of Graph Theory.	Course memento, exercice examples and exercices solving.	2, [1] - [5]
2	Vocabulary of Graph Theory.	idem	2, [1] - [5]
3	Vocabulary of Graph Theory. Shortest path problems in (di)graphs.	idem	2, [1] - [5]
4	Shortest path problems in (di)graphs.	idem	2, [1] - [5]
5	Connectivity problems in (di)graphs (Menger's, Konig's, Hall's theorems).	idem	2, [1] - [5]
6	Minimum spanning trees problem (Prims's and Kruskal's algorithms). Maximum matchings and minimum	idem	2, [1] - [5]

	edge-covers.		
7	Maximum matching problem. Berge's and Tutte's theorems. Hopcroft Karp algorithm. Network flows.	idem	2, [1] - [5]
8			
9	Network flows. Cuts, augmenting paths. Max flow - min cut theorem. Ford Fulkerson and Edmonds Karp algorithms.	Course memento, laboratory woks examples and individual works.	2, [1] - [5]
10	Preflows (Ahuja Orlin algorithm). Combinatorial applications of flows in networks.	idem	2, [1] - [5]
11	Minimum cost flows. Polynomial-time reductions for graph problems.	idem	2, [1] - [5]
12	Polynomial-time reductions for graph problems. Approaching NP-hard problems.	idem	2, [1] - [5]
13	Planar graphs. Drawing planar graphs.	idem	2, [1] - [5]
14	Tree decomposition and applications.	idem	2, [1] - [5]

8. Course content synchronization with the expectations of the community representatives, professional associations and employers from the program domain

This course aims to accommodate the undergraduate students with the classical results and algorithms of Algorithmic Graph Theory. The course acquisitions will be used throughout the undergraduate studies and beyond.

9. Evaluation

Activity Type	9.1 Evaluation criteria	9.2 Evaluation methods	9.3 The weight of each evaluation form (%)
9.4 Course	The correct handling of notions and results from Algorithmic Graph Theory.	Written tests in the regular session.	variable

9.5 Laboratory	The capacity of proof structuring and solving new problems from Algorithmic Graph Theory.	Problems and exercises for seminars and homeworks.	variable
9.6 Minimal performance standards: to understand and to use the abstract concepts and elementary results from Algorithmic Graph Theory.			

Date	Course and Seminary Teacher	Seminary Teachers
26.09.2025	Lect. Dr. Olariu Emanuel Florentin	Lect. Dr. Olariu Emanuel Florentin Lect. Dr. Diac Flavian Paul

Department Date of Approval

Director of the Department
Conf. Andrei Arusoaie, PhD