SYLLABUS

Graph Algorithms

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş Bolyai University
1.2. Faculty	Faculty of Mathematics and Computer Science
1.3. Department	Department of Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Bachelor
1.6. Study programme/Qualification	Artificial Intelligence
1.7. Form of education	Full Time

2. Information regarding the discipline

2.1. Name of the discipline	**Graph Algorithms**					Discipline code	**MLE5025**
2.2. Course coordinator				Mihai Andrei			
2.3. Seminar coordinator				Mihai Andrei			
2.4. Year of study	1	2.5. Semester	2	2.6. Type of evaluation	C	2.7. Discipline regime	Mandatory

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	**4**	of which: 3.2 course	**2**	3.3 seminar/laboratory/project	**2**
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	**28**

Time allotment for individual study (ID) and self-study activities (SA)	**hours**
Learning using manual, course support, bibliography, course notes (SA)	15
Additional documentation (in libraries, on electronic platforms, field documentation)	15
Preparation for seminars/labs, homework, papers, portfolios and essays	30
Tutorship	2
Evaluations	4
Other activities:	3
3.7. Total individual study hours	**69**
3.8. Total hours per semester	**125**
3.9. Number of ECTS credits	**5**

4. Prerequisites (if necessary)

4.1. curriculum	Data Structures and Algorithms
4.2. competencies	Average skills in analysis and design of algorithms and data structures, including implementing them in a programming language. Basic knowledge of Python programming language.

5. Conditions (if necessary)

5.1. for the course	
5.2. for the seminar /lab activities	Laboratory with computers; with Python developing environment.

6.1. Specific competencies acquired [1]

[1] One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

Professional/essential competencies	• create software • define the process
Transversal competencies	• think analytically

6.2. Learning outcomes

Knowledge	The graduate knows, understands and applies the basic concepts and the fundamental algorithms of Artificial Intelligence and is able to evaluate them based on metrics.
Skills	The graduate is able to evaluate, both quantitatively and qualitatively, the performance of intelligent systems. The graduate is able to design and implement software systems that are using methods of Artificial intelligence and to evaluate their performance.
Responsibility and autonomy:	

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Knowing the graph theoretical concepts and using these concepts in the problem modelling. • Knowing how to implement the graph algorithms in a programming language.
7.2 Specific objective of the discipline	• Analysing the issues around the main topics of graph: connectivity, shortest paths, modelling prerequisites and activity planning, flows, traveling salesman problem, planar graphs

8. Content

8.1 Course	Teaching methods	Remarks
1. Basic graph theory definitions (graph, multigraph, directed graph/multigraph, walk, trail, path), basics on graph representations	Exposure, description, explanation, examples, debate	

2. Pseudocode. Walks. Graph measurements: length, distance, diameter, eccentricity, radius, center. Traversals.	Exposure, description, explanation, examples, debate	
3. Minimum-cost walk in a graph. Dijkstra's algorithm (classic), Uniform cost search algorithm (Dijkstra variant),	Exposure, description, explanation, examples, debate	
4. Heuristics, Best First Search, A* and Greedy Search.	Exposure, description, explanation, examples, debate	
5. Bellman-Ford algorithm. Floyd-Warshall algorithm. Subgraphs. Connectivity, connected and strongly-connected components.	Exposure, description, explanation, examples, debate	
6. Trees and forests. Minimum spanning trees. Kruskal and Prim algorithms.	Exposure, description, explanation, examples, debate	
7. Dependency graphs. Topological sorting algorithms. Matchings. Maximum matching in a bipartite graph algorithm.	Exposure, description, explanation, examples, debate	
8. Eulerian cycle. Planar graphs: Euler's relation, K5 and K3,3 graphs, relations between number of edges and vertices.	Exposure, description, explanation, examples, debate	
9. NP-complete problems. Hamiltonian cycle, Traveling Salesman Problem.	Exposure, description, explanation, examples, debate	
10. Other hard problems: clique, vertex cover, colouring.	Exposure, description, explanation, examples, debate	
11. Transport networks. Maximum flow. Ford-Fulkerson algorithm.	Exposure, description, explanation, examples, debate	
12. Maximum flow of minimum cost.	Exposure, description, explanation, examples, debate	
13. Large graphs analysis.	Large graphs analysis.	
Written examination.	Written examination.	

Bibliography

Santosh Kumar Yadav, Advanced Graph Theory, Springer, 2023.
K. Erciyes, Algebraic Graph Algorithms - A Practical Guide Using Python, Springer, 2021.
T. Toadere, Grafe. Teorie, algoritmi si aplicatii, Editura Albastra, 2009.
K. Erciyes, Guide to Graph Algorithms - Sequential, Parallel and Distributed, Springer, 2018.
Shimon Even and Guy Even, Graph Algorithms 2nd edition, 2012.
Saidur Rahman, Basic Graph Theory, Springer, 2017.
Thomas H. Cormen et. al, Introduction to Algorithms, third edition, 2009, MIT Press.

8.2 Seminar / laboratory	Teaching methods	Remarks
Graph representations	Exposure, description, explanation, examples, debate	
More on graph representations. Iterators and traversals.	Exposure, description, explanation, examples, debate	
Minimum cost walks.	Exposure, description, explanation, examples, debate	
Dependency graphs. Minimum spanning trees. Connectivity.	Exposure, description, explanation, examples, debate	
Eulerian circuits. Homeomorphism. Bipartite matching.	Exposure, description, explanation, examples, debate	
NP-complete problems.	Exposure, description, explanation, examples, debate	
Finishing the lab activity.	Exposure, description, explanation, examples, debate	

Bibliography

Santosh Kumar Yadav, Advanced Graph Theory, Springer, 2023.
K. Erciyes, Algebraic Graph Algorithms - A Practical Guide Using Python, Springer, 2021.
T. Toadere, Grafe. Teorie, algoritmi si aplicatii, Editura Albastra, 2009.
K. Erciyes, Guide to Graph Algorithms - Sequential, Parallel and Distributed, Springer, 2018.
Shimon Even and Guy Even, Graph Algorithms 2nd edition, 2012.

Saidur Rahman, Basic Graph Theory, Springer, 2017.
Thomas H. Cormen et. al, Introduction to Algorithms, third edition, 2009, MIT Press.

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The course respects the IEEE and ACM Curricula Recommendations for Computer Science studies; The course exists in the studying program of all major universities in Romania and abroad

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	- know the basic principles of the domain - apply the course concepts - problem solving	Written exam	60%
10.5 Seminar/laboratory	be able to implement course concepts and algorithms	Verifying the practical works.	40%
10.6 Minimum standard of performance			
• At least grade 5 (from a scale of 1 to 10) at both written exam and laboratory work.			

11. Labels ODD (Sustainable Development Goals)[2]

Not applicable.

Date:
04.04.2025

Signature of course coordinator

Digitally signed by ANDREI MIHAI
Date: 2025.04.10 21:15:15 +03'00'

Signature of seminar coordinator

Digitally signed by ANDREI MIHAI
Date: 2025.04.10 21:17:30 +03'00'

Date of approval:
...

Signature of the head of department

Assoc.prof.phd. Adrian STERCA

[2] Keep only the labels that, according to the *Procedure for applying ODD labels in the academic process*, suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.