SYLLABUS

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**

2. Information regarding the discipline

2.1 Name of the discipline (en) (ro)			**Graph Algorithms / Algoritmica grafelor**				
2.2 Course coordinator			**Andrei Mihai**				
2.3 Seminar coordinator			**Andrei Mihai**				
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**C**	2.7 Type of discipline	**Compulsory**
2.8 Code of the discipline		MLE5025					

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	2
3.4 Total hours in the curriculum	56	Of which: 3.5 course	28	3.6 seminar/laboratory	28

Time allotment:	hours
Learning using manual, course support, bibliography, course notes	20
Additional documentation (in libraries, on electronic platforms, field documentation)	10
Preparation for seminars/labs, homework, papers, portfolios and essays	30
Tutorship	4
Evaluations	5
Other activities:	-

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.

5. Conditions (if necessary)

5.1. for the course	
5.2. for the seminar /lab activities	• Laboratory with computers; high level programming language environment (C++, Java, .NET, python)

6. Specific competencies acquired

Professional competencies	C3.2 Identificarea si explicarea modelelor informatice de baza adecvate domeniului de aplicare C3.3 Utilizarea modelelor si instrumentelor informatice si matematice pentru rezolvarea problemelor specifice domeniului de aplicare C 4.2 Interpretarea de modele matematice şi informatice (formale) C 4.3 Identificarea modelelor si metodelor adecvate pentru rezolvarea unor probleme reale
Transversal competencies	CT1 Aplicarea regulilor de muncă organizată şi eficientă, a unor atitudini responsabile faţă de domeniul didactic-ştiinţific, pentru valorificarea creativă a propriului potenţial, cu respectarea principiilor şi a normelor de etică profesională CT3 Utilizarea unor metode şi tehnici eficiente de învăţare, informare, cercetare şi dezvoltare a capacităţilor de valorificare a cunoştinţelor, de adaptare la cerinţele unei societăţi dinamice și de comunicare în limba română și într-o limbă de circulație internațională

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. Subgraphs. Traversals. Connectivity,	Exposure,	

connected and strongly-connected components. Graph measurements: length, distance, diameter, eccentricity, radius, center.	description, explanation, examples, debate	
3. Minimum-cost walk in a graph. Dijkstra's algorithm (classic), Bellman-Ford algorithm.	Exposure, description, explanation, examples, debate	
4. Uniform cost search algorithm (Dijkstra variant), Floyd-Warshall algorithm. Heuristics, Best First Search, A* and Greedy Search.	Exposure, description, explanation, examples, debate	
5. Trees and forests. Minimum spanning trees. Kruskal and Prim algorithms.	Exposure, description, explanation, examples, debate	
6. Dependency graphs. Topological sorting algorithms and strongly connected components algorithms.	Exposure, description, explanation, examples, debate	
7. Independent and dominating sets. 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.	Exposure, description, explanation, examples, debate	
14. Written examination.	Written exam.	
8.2 Seminar	Teaching methods	Remarks
1. Basic definitions. Graph representations.	Exposure, description,	

	explanation, examples, debate	
2. Iterator, traversals.	Exposure, description, explanation, examples, debate	
3. Minimum cost walk algorithms – Dijkstra, Best First Search, Uniform Cost Search, A*	Exposure, description, explanation, examples, debate	
4. Directed acyclic graphs (DAGs). Minimum spanning tree: Kruskal and Prim algorithms.	Exposure, description, explanation, examples, debate	
5. Planar graphs. Matchings.	Exposure, description, explanation, examples, debate	
6. Hard problems in graph theory.	Exposure, description, explanation, examples, debate	
7. Maximum flow.	Exposure, description, explanation, examples, debate	
Laboratory		
1. Graph representations	Exposure, description, explanation, examples, debate	
2. More on graph representations. Iterators and traversals.	Exposure, description, explanation, examples, debate	
3. Minimum cost walks.	Exposure, description, explanation, examples, debate	
4. Dependency graphs. Minimum spanning trees. Connectivity.	Exposure, description, explanation, examples, debate	
5. Planar graphs. Matchings.	Exposure, description, explanation, examples, debate	
6. NP-complete problems.	Exposure, description, explanation, examples, debate	

7. 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.

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

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

Date	Signature of course coordinator	Signature of seminar coordinator
18-04-2024	.....	.....

Date of approval ..... Signature of the head of department .....