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	Computer Science
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				Lect. PhD. Radu Lupsa			
2.3. Seminar coordinator				Lect. PhD. Radu Lupsa			
2.4. Year of study	1	2.5. Semester	1	2.6. Type of evaluation	C	2.7. Discipline regime	Compulsory

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	**1/1/0**
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)					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 skils 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	Lecture room with videoprojector
5.2. for the seminar /lab activities	Room with videoprojector; for laboratory, computers with IDEs for C++, Python, Java and C#

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	• use of theoretical foundations of computer science as well as of formal models • use of software tools in an interdisciplinary context
Transversal competencies	• application of organized and efficient work rules, of responsible attitudes towards the didactic-scientific field, to bring creative value to own potential, with respect for professional ethics principles and norms • use of efficient methods and techniques to learn, inform, research and develop the abilities to bring value to knowledge, to adapt at the requirements of a dynamical society and to communicate efficiently in Romanian language and in an international language

6.2. Learning outcomes

Knowledge	The graduate has the necessary knowledge for using computers, developing software programs and applications, information processing. The graduate has knowledge related to programming, mathematics, engineering and technology and has the skills to use them to create complex information technology systems.
Skills	The graduate has the necessary skills for computer program design and software systems analysis. The graduate has the ability to apply general rules to specific problems and produce relevant solutions.
Responsibility and autonomy:	The graduate is able to identify complex problems and examine related issues to develop solving options and implement solutions. The graduate is able to combine diverse information to formulate solutions and generate ideas for developing new products and applications.

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 modeling. • Knowing how to implement the graph algorithms in a programming language
7.2 Specific objective of the discipline	• Analyzing the issues around the main topics of graph: connectivity, shortest paths, modeling 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),	Exposure: description, explanation, examples, debate.	

basics on graph representations		
2. In-deep study of possible graph representations	Exposure: description, explanation, examples, debate.	
3. Connectivity and shortest path problems. Depth-first and breadth-first traversal of a graph. Connected and strongly-connected components.	Exposure: description, explanation, examples, debate.	
4. Minimum-cost path in a graph. Dynamic programming approach. Bellman-Ford algorithm.	Exposure: description, explanation, examples, debate.	
5. Dijkstra algorithm . A-star algorithm. Floyd-Warshal algorithm.	Exposure: description, explanation, examples, debate.	
6. Dependency graphs, partial order and topological sorting. Topological sorting algorithms and strongly connected components algorithms.	Exposure: description, explanation, examples, debate.	
7. Activity planning problem and algorithms.	Exposure: description, explanation, examples, debate.	
8. Trees and forests. Minimum spanning trees. Kruskal and Prim algorithms.	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, coloring.	Exposure: description, explanation, examples, debate.	
11. Eulerian cycle. Planar graphs: Euler's relation, K5 and K3,3 graphs, relations between number of edges and vertices.	Exposure: description, explanation, examples, debate.	
12. Transport networks. Maximum flow. Ford-Fulkerson algorithm.	Exposure: description, explanation, examples, debate.	
13. Maximum flow of minimum cost.	Exposure: description, explanation, examples, debate.	
14. Matching problem	Exposure: description, explanation, examples, debate.	
Bibliography 1. CORMEN, LEISERSON, RIVEST: Introducere in algoritmi, Editura Computer Libris Agora, 2000. 2. T. TOADERE: Grafe. Teorie, algoritmi si aplicatii , Ed. Albastra, Cluj-N., 2002 3. KÁSA ZOLTÁN: Combinatiroca cu aplicatii, Presa Universitara Clujeana, 2003. 4. BERGE C., Graphes et hypergraphes, Dunod, Paris 1970. 5. BERGE C., Teoria grafurilor si aplicatiile ei, Ed. Tehnica, 1972 6. http://www.cs.ubbcluj.ro/~rlupsa/edu/grafe/		
8.2 Seminar / laboratory	Teaching methods	Remarks
S1. Modelling real world problems with graphs	Dialogue, debate, examples, guided discovery.	
S2. Graph representation	Dialogue, debate, examples, guided discovery.	
S3. Shortest path. Graph traversal	Dialogue, debate, examples, guided discovery.	
S4. Minimum cost path, dynamic programming, Bellman-Ford algorithm	Dialogue, debate, examples, guided discovery.	
S5. Minimum cost path, Dijkstra, A-star, Floyd-Warshall	Dialogue, debate, examples, guided discovery.	
S6. Directed acyclic graphs; trees	Dialogue, debate, examples, guided discovery.	
S7. Maximum flow and matching problems.	Dialogue, debate, examples, guided discovery.	
L1. Graph representation	Dialogue, debate, examples, guided discovery.	
L2. Graph traversal, minimum length path, connected components	Dialogue, debate, examples, guided discovery.	

L3. Minimum cost paths	Dialogue, debate, examples, guided discovery.	
L4. Directed acyclic graphs.	Dialogue, debate, examples, guided discovery.	
L5 Trees.	Dialogue, debate, examples, guided discovery.	
L6. Hard (NP-complete) problems	Dialogue, debate, examples, guided discovery.	
L7 Finalizing assigned problems.	Dialogue, debate, examples, guided discovery.	

Bibliography

1. KÁSA Z., TARTIA C., TAMBULEA L.: Culegere de probleme de teoria grafelor, Lito. Univ. Cluj-Napoca 1979.
2. CATARANCIUC S., IACOB M.E., TOADERE T., Probleme de teoria grafelor, Lito. Univ. Cluj-Napoca, 1994.
3. TOMESCU I., Probleme de combinatorica si teoria grafurilor. Ed. Did. si Pedag. Bucuresti 1981.
4. KÁSA Z., TARTIA C., TAMBULEA L.: Culegere de probleme de teoria grafelor, Lito. Univ. Cluj-Napoca 1979.
5. CATARANCIUC S., IACOB M.E., TOADERE T., Probleme de teoria grafelor, Lito. Univ. Cluj-Napoca, 1994.
6. TOMESCU I., Probleme de combinatorica si teoria grafurilor. Ed. Did. si Pedag. Bucuresti 1981.
7. http://www.cs.ubbcluj.ro/~rlupsa/edu/grafe/

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 Curriculla 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	• Knowing the basic principles of the domain • Applying the course principles • Problem solving	Written exam	60%
10.5 Seminar/laboratory	Being able to implement course concepts and algorithms	Verifying the practical works.	40%

10.6 Minimum standard of performance

- At least 5 of 7 attendances to seminaries and 6 of 7 attendances to labs.
- At least grade 5 (on a scale of 1 to 10) at both written exam and at the average of laboratory works

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

Not applicable.

Date:
...

Signature of course coordinator

.....

Signature of seminar coordinator

.....

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*.".