Software Engineering Institute			Semester 1. of the curriculum 2025-26-1			
Name of the subject:	Code of the subject:	Credits:	Weekly hours:			
				lec	sem	lab
Algorithms and data structures	NSXAA1EBNF	5	full-time	3	0	3
Responsible person for the subject: László Csink			Classification: associate professor			
Subject lecturer(s): László Csink						
Prerequisites:		NSXSFAEBNF Basics of Software Development				
Way of the assessment:	written					

Course description	
Goal:	The aim of the course is to familiarise students with the basic algorithms for optimisation and problem solving and the most common data structures.
Course description:	Common data structures: linked lists, sorted linked lists, stack, queue, priority queue, hash set, dictionary, graph, binary tree. Binary search tree. B-tree. Heap. Hashing. Graphs and graph algorithms.

Lecture schedule	
Education week	Topic
1.	Backtracking Algorithm. Hamiltonian cycle, Sudoku, 8-queens, Chain of words, The knight's tour problem.
2.	Optimization (Introduction to knapsack problem, recursive, memoization and dynamic solution. Coin change problem. Treasure hunt. Greedy algorithms.)
3.	Linked lists. Traversal. Linear search. Insertion of nodes. Deletion of nodes. Linked list implementation. Simple linked list implementation example in C#. Sorted linked lists.
4.	Special data structures. Stack, queue, priority queue, geaps, dictionary.
5.	Hashing. Simple hashing methods. Handling collision. Password protection. A HashSet application.
6.	Graphs. Basic concepts. Graph as data structure. Maze and Minotaur. Components. Breadth-first Search. Depth-first Search. Dijkstra.
7.	Trees. Binary Search Tree. BST implementation in C#. Tree traversal.
8.	B-Tree. Insertion of a key. Deletion of a key a key. Applications.
9.	Trie. Bipartite graph.
10.	Flow. Ford-Fulkerson algorithm. Maximum bipartite matching.
11.	Bitwise algorithms.
12.	Pattern matching. naïve string matching. rabin-kaep algorithm. String matching using finit automaton.
13.	Text indexing. Suffix tree. Suffix array.
14.	Summary.

Mid-term requirements	
Conditions for obtaining a mid-term grade/signature	Attending the lectures and laboratory sessions at the scheduled times is mandatory. In weeks 8 and 13, the students write assessment midterm tests. Both tests are worth 50 points each.

Assessment schedule	
Education week	Topic
8th lab	Midterm test from the topics presented in the lecture and lab
13th lab	Midterm test from the topics presented in the lecture and lab

14th lab	Replacement of either the first or the second midterm test (optional)

Method used to calculate the *mid-term grade* (to be filled out only for subjects with mid-term grades)

The mid-term grade is determined by the sum of the points obtained by the student. Only students having completed the both midterm tests at least 40% and obtained a total of at least 50 points can have a mid-term grade (signature). If the number of absences of the student exceeds 30% of the total number of lessons, the student will be banned from the course.

Type of the replacement

Type of the replacement of written test/mid-term grade/signature	One of the two midterm exams can be retaken in week 14. If the student did not complete the replacement exam at least 40%, or did not manage to achieve at least 50 points out of the 100 points that can be obtained during the semester, the mid-term grade can be only obtained on the mid-term grade replacement exam (signature exam) in the examination period.

Type of the exam (to be filled out only for subjects with exams)

written exam in the exam period

Calculation of the exam mark (to be filled only for subjects with exams)

Maximum possible points for each subtask will be indicated on the exam sheet. A successful exam must exceed 50 % of points.

Final grade calculation methods:

The final grade is the average of the midterm grade and the written exam grade, provided that they are both above minimum level.

average of midterm grade and written test	final mark
0-49%	1
50-61%	2
62-73%	3
74-85%	4
86-100%	5

References

Obligatory:	• Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms (The MIT Press, 2009) • Practical introductory videos and notes available in Moodle • Jon Skeet: C# In Depth. Fourth Edition, Manning Publications Co., 2019. ISBN 9781617294532
Recommended:	
Other references:	bs