Data Structures and Algorithms

(GEMAK121-B2a, GEMAK117-MA)

School year of 2025/26, 2nd semester

for Computer Science Engineering Bachelor programme
and Logistics Engineering Masters programme

Course load: weekly 2 + 2 (lecture + instruction)

Requirements: signature + exam, 5 credits

lecturer & instructor: Viktória Vadon, PhD

Course requirements

- Exit requirements: signature and (written) exam.
- To obtain signature, you must attend 70% of instructions and must pass both midterms separately (reach 50%).
- Midterms:
  - Theoretical midterm, 13 April: based on lecture material, in particular, the glossary of de_nitions and theorems.
  - Practical midterm, 4 May: based on instructions material, fixed exercise types will appear, as shown in the mock practical midterm.
- It is possible to retake either or both midterms independently in the last week of the semester (11 May class).
- If both midterms are successful, and their weighted average (40% theoretical midterm + 60% practical midterm) is at least a grade 3 (60%), this average will be offered as a final grade and you may skip the exam. You can still get the final grade offered from the midterm retakes!
- If both midterms are successful, but their weighted average is only a 2, you get the signature, but you must take the written exam.
- If even after the retakes, one or both midterms are unsuccessful, you must take a “signature retake exam” during the first two weeks of the exam period. This is basically a second retake of the midterms; you can only get the signature, not a final grade. However, you cannot retake the midterms separately anymore, the signature retake exam will have both definitions & theorems and exercises.
- If you have the signature, but a final grade was not offered (or you did not accept it), you must take the written exam. It will contain definitions & theorems as well as exercises, with a couple more exercise types possible.
- Signups for signature retake and written exams must happen through the Neptun online administration system. University rules govern the number of times you may attempt them, and also whether fees are applicable.
- Grading (used for all evaluations): Below 50%: 1 (insufficient). From 50%: 2 (sufficient), 60%: 3 (medium), 70%: 4 (good), 80%: 5 (excellent).

Sources and reading

- Electronic lecture notes, available from the corresponding e-course in the university's E-Learning system. – Somewhat of a work in progress, may be updated throughout the semester.
    - You will be granted access to the e-course after you sign up for the class in the Neptun system.
    - Make sure you have access to the ...@student.uni-miskolc.hu email address/Microsoft account provided by the university, you will need that and the corresponding password to sign in to the e-learning system!)
- Recommended for additional reading: relevant chapters from *Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein-Introduction to algorithms-McGraw-Hill Science_Engineering_Math (2001)*

Weekly curriculum – spring semester of 2026

Week	Date	Lecture	Instruction
1	9 Feb	Course outline, motivation. Program, algorithm. Data, abstract data, data structure. Data types.	Maths prerequisites. Number systems and conversion.
2	16 Feb	Algorithm representations: pseudocode and flowchart. Time and storage complexity, ordo notation. Recurrence equations, the "master theorem", Fibonacci numbers and Binet's formula.	Number representations: unsigned and signed integer, normal form, floating point numbers.
3	23 Feb	Algorithms in number theory. Greatest common divisor and its properties, solvability of the linear congruence equation. Fermat's little theorem.	Ordo notation. Recursive call tree. Why time complexity matters. Time complexity on examples.
4	2 Mar	Dynamic sets. Linear and binary search. Hash tables. Minimum of an array. The "selection problem" and solution in linear time.	Applications of the "Master theorem". Greatest common divisor algorithms pt 1.
5	9 Mar	Insertion sort. Divide and conquer. Quicksort, merge sort, Batcher's even-odd merge. Runtime theorem of comparison-based sorting.	Congruence, linear congruence equation. Multiplicative inverse. Modular exponentiation.
6	16 Mar	Bubble sort, shell sort. Minimum selection sort, square time sort. Linear time sort: counting sort, radix, bucket sort.	Fermat's (pseudo)prime test, RSA algorithm. Dynamic sets pt. 1 (array, linked list).
7	23 Mar	Introduction to graph theory. Trees and properties. Rooted trees, binary trees. Huffman encoding.	Dynamic sets pt. 2: stack, queue, hash tables (open address).

8	30 Mar	Heap, heap sort. Tree traversal and representation. Binary search trees, balancing.	Maximum, selection problem, partition algorithm, quicksort.
9	6 Apr	Easter, school break	
10	13 Apr	**Theoretical midterm.** Graph algorithms. Breadth first search, depth first search.	Counting sort/binsort, Huffman encoding.
11	20 Apr	Optimization problems on trees. Minimum spanning tree. Kruskal and Prim algorithm. Priority queue. Greedy algorithms. Disjoint sets data structure.	Merge sort, heap, heapsort.
12	27 Apr	Single-source shortest paths: stepwise approximation; Dijkstra algorithm, Bellman-Ford algorithm. Shortest paths in directed acyclic graphs.	Revision
13	4 May	All-pairs shortest paths. Floyd-Warshall algorithm. Directed acyclic graph, topological ordering, shortest paths. Strongly connected components. Complexity classes.	**Practical midterm**
14	11 May	**Theoretical midterm retake**	**Practical midterm retake**