Data Structures and Algorithms
(GEMAK117-MA, GEMAK121-B2a)

for Logistics Engineering Masters programme
and Computer Science Engineering Bachelor programme

Course load: weekly 2 + 2 (lecture + instruction)
Requirements: signature + exam, 5 credits

lecturer & instructor: Viktória Vadon, PhD

School year of 2024/25, 2nd semester

1 Course requirements

- Exit requirements: signature and (written) exam.
- To obtain signature, must attend 70% of instructions and must pass both midterms (reach 50%).
- Midterms:
  - "theoretical" midterm, 14 April: based on lecture material, in particular, the glossary of definitions and theorems.
  - "practical" midterm, 12 May: based on instructions material, fixed excercise 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.
- If both midterms are successful, and their average 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 this offered grade from the midterm retakes!
- If both midterms are successful, but their 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, you must take the written exam. You must sign up for this in the Neptun system, I will make several dates available. The final grade will be determined by this written exam. You can sign up for a retake exam free of charge, and a second retake if you pay a fee through the Neptun system.
- Grading (used for all evaluations): Below 50%: 1 (insufficient). From 50%: 2 (sufficient), 60%: 3 (medium), 70%: 4 (good), 80%: 5 (excellent).

2 Lecture notes

- Electronic lecture notes on my website – work in progress, will be updated from time to time!

Miskolc, February 2, 2025

lecturer Viktória Vadon, PhD
(viktoria.vadon@uni-miskolc.hu)

3 Planned weekly curriculum

	Lecture	**Instruction**
1. 10 Feb	Data, abstract data types, data structure. The algorithm. Number representations.	Number systems. Number data types, floating point numbers.
2. 17 Feb	Pseudocode and flowchart. Qualitative description of an algorithm. Growth rate of a function, ordo notation. Fibonacci numbers, Binet-formula. Recurrence equations, the “master theorem”.	Recursion, stack, recursive call tree. Ordo notation, growth order, computation complexity and computation times.
3. 24 Feb	Algorithms in number theory. Greatest common divisor, Euclidean algorithm and its extension, linear congruence equations. Multiplicative inverse, modular powers, Fermat prime test. RSA algorithm.	Applications of the master theorem: computation complexity of recursive algorithms. Algorithms from number theory part 1: greatest common divisor algorithms.

	Lecture	Instruction
4. 3 March	Representing abstract data structures. Dynamic sets. Array, linked list, stack and queue, and their typical applications. Searching in simple structures: linear and logarithmic search. Hash tables. The "selection problem". Finding minimum and maximum. The "selection problem" in linear time.	Algorithms from number theory part 2: greatest common divisor (more algorithms), solution of the linear congruence equation, multiplicative inverse, modular exponentiation. Fermat prime test.
5. 10 March	Insertion sort. Divide and conquer. Merge sort, quicksort. Running times. Runtime theorem for all comparison sorts. Batcher-type merge sort and the related theorem.	Algorithms from number theory part 3: RSA encryption. Data structures part 1: array and its operations.
6. 17 March	Bubble sort, shell sort. Minimum selection sort, square time sort. Linear time sort: counting sort, radix, bucket sort. Sorting with outer memory storage and optimization.	Data structures part 2: linked list and variations, stack, queue.
7. 24 March	Quick introduction to graph theory. Trees, properties of open trees, tree operations. Rooted trees and their representation, binary trees, heap. Heap sort.	Data structures part 3: hash tables (mostly open address hash tables).
8. 31 March	Priority queue. Greedy algorithms. Huffman code. Disjoint sets. Binary trees, binomial heap. search methods. Binary search trees. Red-black trees.	Minimum (or maximum) of array. The partition algorithm, selection problem in linear time. Sorting algorithms part 1: insertion sort, minimum selection sort, bubble sort, shell sort.
9. 7 April	Graph algorithms. Breadth first search, depth first search. Topological ordering. Strongly connected components.	Sorting algorithms part 2: merge algorithm, merge sort, Batcher's even-odd merge, quicksort, binsort (counting sort), radix sort.
10. 14 April	**Theoretical midterm, 45 minutes.** Optimization problems on trees. Minimum spanning tree. Kruskal and Prim algorithm.	Heap data structure, heap sort. Huffman encoding.

	Lecture	**Instruction**
11. ~~21 April~~	~~Easter.~~	~~Easter.~~
12. ~~28 April~~	~~School holiday.~~	~~School holiday.~~
13. 5 May	Single-source shortest paths, stepwise approximation. Dijkstra algorithm. Bellman-Ford algorithm. Shortest paths in directed acyclical graphs.	Revision, practicing midterm exercises.
14. 12 May	All-pairs shortest paths. Floyd-Warshall algorithm. Transitive closure of graphs, Warshall algorithm. Dynamic programming: principle and application to finite matrix products. Algorithmic solvability, relation of P and NP classes, examples of P and NP problems.	**Practical midterm, 90 minutes.**
15. 19 May	**Theoretical midterm retake.** Possibly revision and practicing for the exam.	**Practical midterm retake.**