

Course Specification Document

Title	Fundamentals of Algorithms and Data Structures
--------------	--

Credits	3.5 ECTS
----------------	----------

Aims	This course aims to introduce the student to algorithm structure and analysis, diverse search and sorting algorithms, and abstract data structures (stacks, queues, lists, trees), alongside various graph problems.
-------------	--

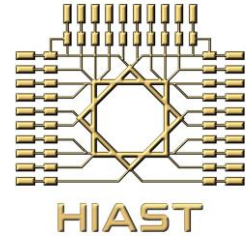
Intended learning outcomes

On successful completion of this course, the student will be able to:

- Understand algorithm structures and become acquainted with methods for analyzing algorithms.
- Identify sorting, ordering, and comparison algorithms and distinguish between them.
- Grasp abstract data structures, understand their benefits, and distinguish the differences between them.
- Understand the structure of graphs and interpret problem-solving methodologies for graph-related issues.
- Grasp algorithms used to tackle traditional graph problems (such as finding the shortest path in a graph, identifying minimum spanning trees).
- Compute the time complexity of algorithms and prove their correctness.
- Implement abstract data structures and proficiently execute fundamental operations on them.

Syllabus

- **Problem-solving using algorithms:** Definition of algorithms, components of algorithms, Gradual Algorithm design, pseudocode.
- **Algorithms correctness and complexity:** Algorithms correctness, number of operations in an algorithm - measuring execution time, average and worst-case algorithm complexity, algorithm optimality.
- **Search and sorting algorithms:** Sequential search implementation, complexity, binary search implementation, complexity, sorting - stable sorting, sorting algorithms and analysis: insertion sort, selection sort, bubble sort, quick sort, merge sort.
- **Abstract data structures "List":** "List" data structure and its applications, static representation of the "List" and its implementation, dynamic representation of the "List" and its implementation, "List" operations: add, search, delete...
- **Abstract data structures "Stack":** "Stack" data structure and its applications, static representation of the "Stack" and its implementation, dynamic representation of the "Stack" and its implementation, "Stack" operations: add, search, delete,



- **Abstract data structures "Queue":** "Queue" data structure and its applications, static representation of the "Queue" and its implementation, dynamic representation of the "Queue" and its implementation, "Queue" operations: add, search, delete...
- **Abstract data structures "Tree":** "Tree" data structure and its applications, static representation of the "Tree" and its implementation, dynamic representation of the "Tree" and its implementation, binary trees and their applications, balanced trees.
- **Abstract data structures "Graph":** Definition of the abstract data structure "Graph" and its applications and types, graph representation, graph traversal algorithms, finding minimum spanning tree algorithms, shortest path algorithms.