UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

SYLLABUS DEL CORSO

Algoritmi e Strutture Dati

2627-1-E3102Q106

Aims

The aim of the course is to teach to design, evaluate and implement efficient algorithms, making use of the most appropriate progamming techniques and data structures.

Knowledge and Understanding

- Knowledge of basic concepts related to computational problems
- Knowledge of basic techniques to prove the correctness of an algorithm
- Knowledge of how recursive and divide-and-conquer programming paradigms work
- Knowledge of how sorting algorithms work
- Knowledge of techniques to evaluate the efficiency of algorithms and data structures
- Knowledge of how different fundamental data structures work and their characteristics

Applying Knowledge and Understanding

- Ability to analyze the formal correctness of iterative and recursive algorithms
- Ability to apply recursive and divide-and-conquer paradigms to solve new computational problems
- Ability to apply techniques to evaluate the efficiency of iterative, recursive, and divide-and-conquer algorithms
- Ability to simulate how complex algorithms and fundamental data structures work on specific examples
- Ability to correctly use fundamental data structures to solve new computational problems

Making Judgments

- Ability to choose the most efficient and/or most suitable algorithm to solve specific new computational problems
- Ability to choose the most efficient and/or most suitable data structure to solve specific new computational problems

Communication Skills

- Ability to correctly use pseudocode to formally and clearly present an algorithm
- Ability to understand and use the correct terminology to describe or understand the characteristics of an algorithm or a data structure

Learning Skills

- Ability to search for and critically select a new algorithm that is more suitable for solving a new computational problem
- Ability to search for and critically select a new data structure that is more suitable for solving a new computational problem

Contents

Basic techniques to develop algorithms and to analyse their efficiency. Introduction to the use of fundamental data structures.

Detailed program

- Introduction and basic definitions: algorithm, problem, instance.
- Computational complexity analysis of algorithms.
- Recursive programming and Divide-and-Conquer programming technique: Mergesort and Quicksort.
- Time complexity for recursive algorithms: recursive equations.
- Linear time sorting algorithms.
- Basic data structures: arrays, linked lists, stacks, queues.
- Binary trees and Search Binary Trees.
- Heap and priority queues. Heapsort.
- Graphs and graph representation.
- Traversing algorithms for graphs: BFS and DFS

Prerequisites

Basics of Computer Programming and Calculus.

Teaching form

Theoretical lectures, exercises, and practical implementation of proposed algorithms. Further exercises are available through the textbook.

The course is taught in Italian. All activities are in-person: 32 hours in unidirectional mode (2 or 3 hours per lesson), 44 hours in interactive mode, of which 20 hours (2 or 3 hours per lesson) for exercises and 24 (4 hours per lesson) for laboratory.

Textbook and teaching resource

T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, Mit Press ed.

Semester

Second semester

Assessment method

Assessment of learning outcomes is carried out through an **individual written examination** aimed at verifying the achievement of the course learning objectives related to the knowledge of algorithms and data structures, the ability to analyze their correctness and efficiency, and the ability to apply such knowledge to the solution of computational problems.

The written examination is graded on a **30-point scale** and includes:

- Exercises requiring the design and development of recursive or iterative algorithms for problems not directly addressed during the course, aimed at assessing the ability to apply knowledge and methodologies to problem solving (*Dublin Descriptor 2*);
- Exercises involving the simulation and analysis of the behavior of algorithms and data structures covered during the course, aimed at assessing the understanding of their operating principles and the ability to analyze their correctness and efficiency (*Dublin Descriptors 1 and 2*);
- Open-ended questions aimed at assessing knowledge of the fundamental concepts, definitions, and properties of algorithms and data structures, as well as the ability to use the appropriate disciplinary terminology (*Dublin Descriptors 1 and 4*).

The theory questions contribute up to **8 points** overall.

As an alternative to the final written examination, **two midterm examinations** are offered to first-year students. These are held halfway through the course and during the first examination session in June. Each midterm has the same structure as the full written examination and assesses the learning outcomes related to the topics covered up to that point.

Each midterm examination is graded on a **30-point scale** and is considered passed if a score of at least **16/30** is achieved. If both midterms are passed, the final grade is obtained as the arithmetic mean of the two scores, rounded up to the nearest integer. The examination is considered passed if the resulting final grade is at least **18/30**.

Students may retake **only one** of the two midterm examinations during the July examination session in case of absence, failure, or to improve the previously obtained result.

Assessment Criteria

The evaluation of written examinations is based on the following criteria:

- Formal correctness of the proposed solutions;
- Ability to identify and apply appropriate algorithms and data structures;
- Ability to analyze correctness, computational complexity, and efficiency of the proposed solutions;
- Completeness and accuracy of the answers;
- Clarity of presentation and appropriate use of technical terminology and pseudocode.

The grading scale is interpreted as follows:

- 18-21/30: achievement of the essential learning outcomes;
- 22-25/30: good understanding of the topics and correct application of the main techniques;
- 26-29/30: thorough knowledge of the topics and ability to solve complex problems independently;
- 30/30: complete mastery of the topics, rigorous correctness of the solutions, and full achievement of the expected communication skills.

The grade of 30 cum laude is awarded for outstanding performances demonstrating full achievement of the learning objectives, together with exceptional rigor, completeness, and clarity of presentation. A grade of 30 cum laude obtained in a midterm examination is counted as 31 points for the purpose of calculating the final average.

Office hours

By appointment

Sustainable Development Goals