UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Algoritmi e Strutture Dati

2526-1-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

Teaching form

Theoretical lectures, exercises, and practical implementation of proposed algorithms. Further exercises are available online, through an E-learning website.

The course is taught in Italian. All activities are in-person: 32 hours in unidirectional mode, 44 hours (20 for exercises and 24 for laboratory) in interactive mode.

Textbook and teaching resource

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

Further material available through the e-learning website.

Semester

Second semester

Assessment method

Written examination, maximum mark 30/30. The exam consists of

- Exercises to be solved by developing an algorithm to solve a given computing problem
- Simulations on specific inputs for algorithms illustrated during the lectures
- Open questions concerning theoretical aspects discussed during the lectures

Score for theoretical questions: 8 points (in total)

Up to 2 more points can be given in case of extremely good exercises.

The exam can be substituted by two intermediate exams, each evaluating some of the subjects covered during the course.

Split examination:

Written examination can be replaced by two split short exams, held during mid course break and at the end of the course. Such possibility is only allowed to first-year students. Each short exam consist of:

- An exercise to be solved by developing an algorithm to solve a given computing problem
- A simulation on specific inputs of algorithms illustrated during the lectures
- Open questions concerning theoretical aspects discussed during the lectures

The maximum score for each short exam is 15/15. Final mark is obtained by summing the marks of the two short exams. Students can repeat one of the two short exams (in case of non sufficient result, or to improve the result) during the exam in july.

Up to 2 more points can be given (considering both short exams) in case of extremely good exercises.

Office hours

By appointment

Sustainable Development Goals