UNIVERSITY OF BASILICATA STUDIES
DEPARTMENT OF MATHEMATICS, INFORMATICS AND ECONOMICS

COURSE: Algorithms and Data Structures I			
ACADEMIC YEAR: 2019/20			
TYPE OF EDUCATIONAL ACTIVITY: Basic			
TEACHER: Erra Ugo			
e-mail: ugo.erra@unibas.it		website:	
phone: +39 0971 205869		mobile (optional):	
Language: Italian			
ECTS: 6	n. of hours: 48	Campus: Potenza Dept./School: DIMIE Program:	Semester: I

EDUCATIONAL GOALS AND EXPECTED LEARNING OUTCOMES

The main objective of the course is to provide the notion of algorithm and data structures. Characterize the data to be processed, organizing, and structuring to facilitate their use by the algorithms. Designing correct and efficient algorithms through the examination of different paradigms.

- **Knowledge and comprehension**: the student must demonstrate to know and understand the concept of the algorithm, to know how to prove its correctness, and calculate its efficiency. The student must also have understood the main algorithms and basic data structures to design new and efficient algorithms;
- **Ability to apply knowledge and understanding**: at the end of the course the student will acquire a library of algorithms and data structures that he can use to design new algorithms but also during the programming of an algorithm. Understanding the topics will provide the ability to design new algorithms but also deepening towards more sophisticated algorithms;
- **Autonomy of judgment**: the student must be able to design and evaluate an algorithm in terms of correctness and efficiency;
- **Communication skills**: the student must have the ability to present clearly, using, if necessary, a language that can be understood even by non-experts, the functioning of an algorithm or a data structure using the tools illustrated in the lesson to demonstrate its correctness and efficiency;
- **Learning ability**: the student must be able to consult texts and scientific articles independently to extend the basic knowledge acquired during the course.

PRE-REQUIREMENTS

Procedural programming (dynamic data structure, pointers, event-driven programming). In particular, good knowledge of a procedural language as for instance C language.

SYLLABUS

The Role of Algorithms in Computing (2 hrs lessons): algorithms, algorithms as a technology;

Growth of Functions (2 hrs lessons): asymptotic notation, standard notations and common functions;

Elementary Data Structures (2 hrs lessons): stacks and queues, linked lists, implementing pointers and objects;

Divide-and-Conquer (2 hrs lessons): the substitution method for solving recurrences, the recursion-tree method for solving recurrences, the master method for solving recurrences;

Probabilistic Analysis and Randomized Algorithms (4 hrs lessons): the hiring problem, indicator random variables, randomized algorithms, probabilistic analysis and further uses of indicator random variables;

Heapsort (4 hrs lessons): heaps, maintaining the heap property, building a heap, the heapsort algorithm, priority queues;

Quicksort (4 hrs lessons): description of quicksort, performance of quicksort, randomized version of quicksort, analysis of quicksort;

Sorting in Linear Time (2 hrs lessons): lower bounds for sorting, counting sort, radix sort, bucket sort;

Hash Tables (4 hrs lessons): direct-address tables, hash tables, hash functions, open addressing;

Binary Search Trees (4 hrs lessons): what is a binary search tree?, querying a binary search tree, insertion and deletion;

Dynamic Programming (6 hrs lessons): rod cutting, matrix-chain multiplication, elements of dynamic programming, longest common subsequence;

Greedy Algorithms (6 hrs lessons): an activity-selection problem, elements of the greedy strategy, huffman codes;

Graph Algorithms (6 hrs lessons): fundamental concepts of graph theory, visit algorithms, topological sorting.

TEACHING METHODS

During the lesson will be illustrated all the content of the syllabus. The lesson will be divided into two parts. In the first part, will be presented the theoretical aspects while in the second part will be presented practical examples. In such a way, the student is invited to use immediately the concepts introduced in class. Also, during the course will be released self-assessment tests. Such tests are heterogeneous with difficulty, from simple definitions to a request for a solution. Regular participation in the lesson is strongly recommended.

EVALUATION METHODS

The examination consists of a written evaluation with six open questions of a theoretical nature or resolution of exercises. There is a maximum of 5 points per assessment. After passing the written evaluation, it is possible to ask for an addition to the written evaluation through an oral evaluation with in-depth questions about the entire course program. Alternatively, there will be two intermediate written evaluations during the course. The first written evaluation will be held during the teaching break and will cover only the topics of the first part of the course. The second evaluation at the end of the course will cover the remaining part of the course. In both evaluations, there are six exercises where for each exercise there is a maximum of 3 points.

TEXTBOOKS AND ON-LINE EDUCATIONAL MATERIAL

Introduction to Algorithms and Data Structures 3/ed. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

INTERACTION WITH STUDENTS

At the beginning of the course, the teacher describes the course's objectives, syllabus, and validation tests, indicating where to find the teaching material. The reception time is set for Tuesday and Wednesday from 10:30 a.m. to 12:30 p.m. in the teacher's office. In addition to the weekly reception time, the teacher is available with students by e-mail or at the end of the lesson.

EXAMINATION SESSIONS (FORECAST)[1]

4/2/2020, 18/2/2020, 6/5/2020, 1/7/2020, 15/7/2020, 23/9/2020, 16/12/2020

SEMINARS BY EXTERNAL EXPERTS YES □ NO x

FURTHER INFORMATION

[1] Subject to possible changes: check the web site of the Teacher or the Department/School for updates.