
Ano Letivo 2018-19

Unidade Curricular ALGORITMOS E ESTRUTURAS DE DADOS

Cursos ENGENHARIA INFORMÁTICA (1.º ciclo)

Unidade Orgânica Faculdade de Ciências e Tecnologia

Código da Unidade Curricular 14781051

Área Científica CIÊNCIA DE COMPUTADORES

Sigla

Línguas de Aprendizagem Português

Modalidade de ensino Ensino presencial

Docente Responsável Pedro João Valente Dias Guerreiro

DOCENTE	TIPO DE AULA	TURMAS	TOTAL HORAS DE CONTACTO (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1	30T; 30PL
Docente A Contratar FCT 1	PL	PL2; PL3	60PL

* Para turmas lecionadas conjuntamente, apenas é contabilizada a carga horária de uma delas.

ANO	PERÍODO DE FUNCIONAMENTO*	HORAS DE CONTACTO	HORAS TOTAIS DE TRABALHO	ECTS
2º	S2	30T; 30PL	168	6

* A-Anual;S-Semestral;Q-Quadrimestral;T-Trimestral

Precedências

Sem precedências

Conhecimentos Prévios recomendados

Idealmente, os alunos terão realizado com êxito as cadeiras de programação antecedentes: Programação Imperativa, Laboratório de Programação e Programação Orientada por Objetos.

Objetivos de aprendizagem (conhecimentos, aptidões e competências)

No fim desta cadeira, todos os alunos deverão ser capazes de:

- Compreender as estruturas de dados fundamentais: vetores, listas, filas, pilhas, árvores, tabelas de dispersão, ?union-find?.
- Compreender os principais algoritmos de ordenação.
- Compreender os principais algoritmos sobre grafos.
- Compreender os principais algoritmos sobre cadeias de caracteres e vetores de cadeias de caracteres.
- Saber utilizar bibliotecas que implementam essas estruturas de dados e esses algoritmos.
- Programar essas estruturas de dados, adaptando-as a necessidades supervenientes e acrescentando novas funcionalidades.
- Distinguir as características de complexidade dos principais algoritmos.
- Compreender e saber aplicar as ideias fundamentais da programação dinâmica.
- Apreçar melhor a problemática do desenvolvimento de software e a importância da qualidade do software.

Além disto, os alunos terão reforçado as suas competências gerais de programação.

Conteúdos programáticos

- Conceitos fundamentais
 - Modelo de programação
 - Sacos, pilhas e filas
 - *Union-Find*
 - Análise de algoritmos
- Ordenação
 - Algoritmos elementares
 - Mergesort, quicksort
 - Filas com prioridade
- Busca
 - Árvores binárias de busca
 - Árvores equilibradas
 - Tabelas de dispersão
- Grafos
 - Busca em profundidade
 - Busca em largura
 - Árvores de cobertura
 - Caminho mais curto
- Cadeias de caracteres
 - Busca de subcadeias
 - Compressão de dados
- Estratégias programativas
 - Divisão-conquista
 - Algoritmos vorazes ("gananciosos")
 - Programação dinâmica
- Intratabilidade

Metodologias de ensino (avaliação incluída)

Nas aulas teóricas o professor faz a exposição da matéria usando o quadro e o computador para apresentar os conteúdos e fazer demonstrações.

Nas aulas práticas os alunos trabalham na resolução de exercícios.

Critérios de Avaliação

* trabalhos práticos (30%)

* exame final (70%)

É necessário uma nota não inferior a 7,0 valores na avaliação da componente prática para serem admitidos a exame final. Arredondamentos só na nota final.

Os trabalhos práticos são feitos individualmente ou em grupo de 2 pessoas. Cada trabalho terá o seu próprio prazo de entrega que será avisado atempadamente durante o decurso da disciplina. A nota prática do grupo é convertida numa nota prática individual no momento da discussão dos trabalhos (que ocorrerá na última semana de aulas). A nota prática é, portanto, individual, e depende do desempenho de cada elemento do grupo na discussão.

Bibliografia principal

Introduction to Algorithms, 3rd Edition
Cormen, Leiserson, Rivest, and Stein. MIT Press, 2009.

Academic Year 2018-19

Course unit ALGORITHMS AND DATA STRUCTURES

Courses INFORMATICS (COMPUTER SCIENCE) (1st Cycle)

Faculty / School Faculdade de Ciências e Tecnologia

Main Scientific Area CIÊNCIA DE COMPUTADORES

Acronym

Language of instruction Portuguese

Teaching/Learning modality face to face learning

Coordinating teacher Pedro João Valente Dias Guerreiro

Teaching staff	Type	Classes	Hours (*)
Pedro João Valente Dias Guerreiro	PL; T	T1; PL1	30T; 30PL
Docente A Contratar FCT 1	PL	PL2; PL3	60PL

* For classes taught jointly, it is only accounted the workload of one.

Contact hours

T	TP	PL	TC	S	E	OT	O	Total
30	0	30	0	0	0	0	0	168

T - Theoretical; TP - Theoretical and practical ; PL - Practical and laboratorial; TC - Field Work; S - Seminar; E - Training; OT - Tutorial; O - Other

Pre-requisites

no pre-requisites

Prior knowledge and skills

Ideally, students will have successfully completed the previous programming courses in the curriculum: Imperative Programming, Programming Lab and Object Oriented Programming.

The students intended learning outcomes (knowledge, skills and competences)

At the end of this course, students should be able to:

- Understand the fundamental data structures: vectors, lists, queues, stacks, trees, graphs, union-find.
- Understand the main sorting algorithms.
- Understand the main algorithms on graphs.
- Understand the main algorithms on strings and on arrays of strings.
- Know how to use libraries that implement these data structures and these algorithms.
- Program these data structures, adapting them to future needs and adding new functionalities.
- Distinguish the complexity characteristics of the main algorithms.
- Understand and apply the fundamental ideas of dynamic programming.
- Appreciate the software development life cycle and the importance of software quality.

In addition, students will have strengthened their overall programming skills.

Syllabus

- Fundamental concepts
 - Programming model
 - Bags, stacks and queues
 - Union-Find
 - Analysis of algorithms
- Sorting
 - Elementary algorithms
 - Mergesort, quicksort
 - Priority queues
- Searching
 - Binary search trees
 - Balanced trees
 - Hash tables
- Graphs
 - Depth-first search
 - Breadth-first search
 - Spanning trees
 - Shortest paths
- Strings
 - Substring search
 - Data compression
- Programming strategies
 - Divide and conquer
 - Greedy Algorithms
 - Dynamic programming
- Intractability

Teaching methodologies (including evaluation)

In the main T lectures, the course materials are explained to students along with illustrative examples. In the P lectures students have hands-on experience on the course materials doing exercises.

Grading:

practical component (30%)
final exam (70%)

Students need a grade greater or equal to 7,0 in the practical component to be admitted to the final exam.

The practical component consists mostly of programming assignments and has to be done individually or in a group of a maximum of 2 people. Each assignment will have its own deadline which will be announced during the course. The grade for the practical component of the group will be converted into a student grade upon the mandatory individual discussion that will take place in the last week of lectures of the semester.

Main Bibliography

Introduction to Algorithms, 3rd Edition
Cormen, Leiserson, Rivest, and Stein. MIT Press, 2009.