

TeSP - Tecnologia e Programação em Sistemas de Informação

Técnico Superior Profissional

Plano: Aviso nº 909/2016 - 27/01/2016

Ficha da Unidade Curricular: Algoritmos

ECTS: 2; Horas - Totais: 54.0, Contacto e Tipologia, TP:37.50;

Ano|Semestre: 2|S1; Ramo: Tronco comum;

Tipo: Obrigatória; Interação: ; Código: 602421

Área de educação e formação: Matemática

Docente Responsável

Maria Manuela Morgado Fernandes Oliveira

Docente e horas de contacto

Maria Manuela Morgado Fernandes Oliveira

Equiparado Assistente 2º Triénio, TP: 37.5;

Objetivos de Aprendizagem

Dar aos formandos a capacidade analítica que lhes permita construir, desenvolver e adaptar algoritmos de modo a serem capazes de conceber soluções lógicas para problemas surgidos nos mais diversos âmbitos, nomeadamente na informática, e contribuir assim na conceção e construção de *software*, individualmente ou integrados em equipas.

Objetivos de Aprendizagem (detalhado)

Aspectos introdutórios. Funções. Algoritmos de ordenação e de pesquisa. Pesquisa linear e pesquisa binária. Algoritmos sobre Grafos e Digrafos. Extensões: noções básicas sobre heurísticas.

Conteúdos Programáticos

Aspectos introdutórios: Breve introdução ao *Octave* em ambiente *Windows*; dados, estruturas de dados e estruturas de controle; vetores e matrizes; funções recursivas e não recursivas. **Algoritmos de ordenação e de pesquisa:** *Bubblesort*, ordenação por seleção, ordenação por inserção, *shellsort*, *mergesort* e *quicksort*. Pesquisa linear e pesquisa binária. **Algoritmos sobre Grafos:** definição e propriedades fundamentais dos grafos; estruturas de dados para representação, armazenagem e manipulação de grafos; construção de caminhos e ciclos em grafos; grafos conexos; árvores; extensão aos digrafos e a redes. **Aplicações:** algoritmo DFS para a construção de uma árvore geradora de um grafo conexo; algoritmo para a construção de um ciclo euleriano; o problema da determinação de uma árvore geradora de custo mínimo; algoritmos de Kruskal e de Prim; o problema da determinação de um caminho de custo mínimo numa rede: algoritmos de Dijkstra e de Floyd-Marshall; Problema do fluxo máximo numa rede: algoritmo de Ford-Fulkerson. **Extensões:** noções básicas sobre heurísticas; aplicação ao problema do caixeiro-viajante.

[Handwritten signature]
[Handwritten signature]

Conteúdos Programáticos (detalhado)

Aspectos introdutórios: Breve introdução ao *Octave* em ambiente *Windows*; dados, estruturas de dados e estruturas de controle; vetores e matrizes; funções recursivas e não recursivas. **Algoritmos de ordenação e de pesquisa:** *Bubblesort*, ordenação por seleção, ordenação por inserção, *shellsort*, *mergesort* e *quicksort*. Pesquisa linear e pesquisa binária. **Algoritmos sobre Grafos:** definição e propriedades fundamentais dos grafos; estruturas de dados para representação, armazenagem e manipulação de grafos; construção de caminhos e ciclos em grafos; grafos conexos; árvores; extensão aos dígrafos e a redes. **Aplicações:** algoritmo DFS para a construção de uma árvore geradora de um grafo conexo; algoritmo para a construção de um ciclo euleriano; o problema da determinação de uma árvore geradora de custo mínimo: algoritmos de Kruskal e de Prim; o problema da determinação de um caminho de custo mínimo numa rede: algoritmos de Dijkstra e de Floyd-Marshall; Problema do fluxo máximo numa rede: algoritmo de Ford-Fulkerson. **Extensões:** noções básicas sobre heurísticas; aplicação ao problema do caixeiro-viajante.

Metodologias de avaliação

Avaliação Contínua: Realização de duas provas escrita e dois projetos intercalares.

Cada projeto intercalar vale 20 valores com nota mínima de 6 valores. Cada prova escrita é cotada de 0 a 20 valores e deverão ter nota mínima de 6 valores. A nota final será dada pela média aritmética das quatro provas.

Avaliação Final: Os alunos admitidos a exame, ou os dispensados que pretendam melhorar a sua nota, podem fazer o exame final, que consiste numa prova escrita com vertente computacional sobre toda a matéria lecionada. Há lugar a um exame de recurso, para os alunos que reprovarem na época normal. As notas dos trabalhos poderão ou não transitar para exame, conforme escolha do aluno. Dando-se ainda a hipótese de melhorar, uma única vez, os trabalhos até à data do exame.

Software utilizado em aula

Octave em ambiente *Windows* ou outro à escolha do aluno.

Estágio

Não se aplica.

Bibliografia recomendada

Apontamentos fornecidos pelo docente, e disponibilizados aos alunos através da plataforma de *e-learning* do IPT; K. Rosen, *Discrete Mathematics and its Applications*, McGraw-Hill, 2003; N. Wirth, *Algorithms + Data Structures = Programs*, Prentice-Hall, 1976; A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

Coerência dos conteúdos programáticos com os objetivos

Nos nossos dias, a recolha, organização, armazenamento e pesquisa de dados é um tema da maior importância, justificando pois a abordagem dos temas: estruturas de dados, algoritmos de ordenação e pesquisa. Na era das redes sociais, redes de telecomunicações, das grandes redes de distribuição e recolha de materiais e/ou produtos, das enormes rotas comerciais, de transportes ou elétricas, é essencial perceber que é possível representar estas estruturas através de grafos e dígrafos. Para além disso, é incontornável a necessidade de entender e saber implementar, algoritmos como o BFS, DFS, Kruskal, Prim, Dijkstra de Floyd-Marshall ou Ford-Fulkerson, uma vez que estes dão frequentemente resposta a problemas reais. Para terminar, é preciso ter a noção, que embora a evolução dos computadores e das ciências da computação ser abismal, há ainda problemas cuja resolução exata é pouco provável em tempo útil tornando a utilização de heurísticas crucial para obtenção de uma solução, embora sem garantia que seja a melhor possível. Desta forma, é introduzido o conceito de heurística e é dado um exemplo da sua utilização, no caso do Problema do Caixeiro Viajante.

Metodologias de ensino

As aulas decorrem predominantemente em ambiente computacional, permitindo aos alunos, para além de discutirem e decidirem qual a sequência de procedimentos a efetuar para atingir um determinado objetivo, constatarem de imediato se essa sequência funciona. São abordados diversos algoritmos com grande aplicação prática e o aluno, para além da sua compreensão teórica, deverá proceder à sua implementação.

Coerência das metodologias de ensino com os objetivos

A metodologia de ensino reforça o conhecimento, compreensão e capacidade de implementação de algoritmos. Contribui ainda, para a flexibilização do raciocínio e desenvolvimento da capacidade de conceção de soluções lógicas na resolução de problemas surgidos no âmbito da informática, e mesmo noutros âmbitos, como por exemplo, a recolha de lixos, definição de rotas de distribuição, instalação de redes de esgotos, definição de redes de telecomunicações, etc. A discussão fomentada em sala de aula permite desenvolver um sentido crítico e espírito de equipa, tão necessário a profissionais, e mesmo académicos, nas áreas da computação. Esta prática e o conhecimento teórico adquirido ajudará na construção de *software*, de forma individual ou integrada em equipas.

Língua de ensino

Português

Pré requisitos

Não se aplica

Programas Opcionais recomendados

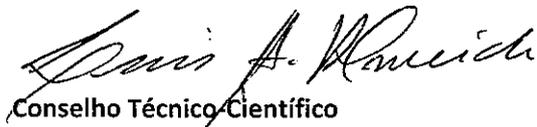
Não se aplica

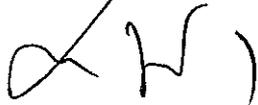
Observações

Nada a observar


Docente Responsável

Diretor de Curso, Comissão de Curso


Conselho Técnico Científico



Homologado pelo C.T.C.
Acta n.º 12 Data 7/1/2018
