

CONCEPÇÃO E ANÁLISE DE ALGORITMOS (CAL)

2019-2020 (2S)

**ROSALDO ROSSETTI, LILIANA FERREIRA, HENRIQUE
LOPES CARDOSO, FRANCISCO ANDRADE**



DOCENTES



Rosaldo Rossetti
(rossetti@fe.up.pt)



Liliana Ferreira
(lsferreira@fe.up.pt)



Henrique Lopes Cardoso
(hlc@fe.up.pt)



Francisco Andrade
(franciscora@fe.up.pt)

OBJETIVOS

Gerais

- Complementar e aprofundar conhecimentos de “Prog.” e “AED”
- Introduzir técnicas de concepção e implementação de algoritmos eficientes para resolução de diferentes tipos de problemas
- Analisar e avaliar as soluções concebidas

Competências a adquirir

- Conhecer e saber aplicar algoritmos eficientes em grafos, conjuntos e cadeias de caracteres
- Conhecer e saber aplicar técnicas genéricas de concepção e análise de algoritmos
- Conhecer alguns problemas intratáveis e algoritmos que fornecem soluções aproximadas para alguns deles

Resultados da aprendizagem

- **caracterizar um problema, formalizá-lo, conceber algoritmos eficientes** para solucioná-lo, e **avaliar a solução concebida**

PROGRAMA

- **Técnicas de concepção de algoritmos:** divisão e conquista; algoritmos gananciosos; programação dinâmica; algoritmos de retrocesso; algoritmos probabilísticos
- **Formalização** de problemas; **representação** de algoritmos; **análise** da sua **complexidade** (temporal e espacial); **verificação** da **correção** dos algoritmos
- **Estruturas de dados avançadas:** filas de prioridade com alteração de prioridade; conjuntos disjuntos
- **Algoritmos** eficientes em **grafos**: ordenação topológica; caminho mais curto; árvore de expansão mínima; fluxo máximo e fluxo máximo de custo mínimo em redes de transporte; circuito de Euler e problema do carteiro chinês
- **Algoritmos** em “**strings**”: pesquisa exacta e aproximada; “substring” comum mais comprida; compressão de texto
- **Problemas intratáveis:** teoria dos problemas NP-completos

BIBLIOGRAFIA

Principal

- T. Cormen; C. Leiserson; R. Rivest; C. Stein. **Introduction to Algorithms**. Cambridge, MA: MIT Press, 2009.
- M.A. Weiss. **Data Structures and Algorithm Analysis in C++**, 3/E. New York, NY: Addison Wesley, 2007.
- S. Skiena. **The Algorithm Design Manual**. Berlin: Springer, 2008.

Outras referências

- R. Sedgewick. **Algorithms in C++ Part 5: Graph Algorithms**, 3/E. New York, NY: Addison Wesley, 2002.

Material de apoio

- Moodle da disciplina!

MÉTODO DE ENSINO

As aulas teóricas são usadas para a exposição formal da matéria, acompanhada da apresentação de exemplos e sua discussão

As aulas práticas são usadas para a resolução de exercícios e desenvolvimento de pequenos programas em C++, para testar os algoritmos desenvolvidos

Os estudantes também deverão realizar trabalhos práticos, em grupos de 3 (três) estudantes. Apesar de realizados em grupo, a avaliação dos trabalhos é individual!

Avaliação individual, por exame final e pela observação da assiduidade, participação e desempenho nas aulas práticas

AVALIAÇÃO

Avaliação distribuída com exame final

Frequência (CD)

- Frequência mínima às aulas práticas de laboratório (75%)
- Trabalho prático de grupo (CG) (3 estudantes/grupo)
 - Parte I: Formalização (25%) ≥ 8.0 valores
 - Parte II: Implementação e Análise (75%) ≥ 8.0 valores
- Nota mínima em cada parte do trabalho: 40%

Exame final (EF) ≥ 8.0 valores

- Consulta condicionada; duração 2 horas

Avaliação Final

- $AF = 0,6 \times EF + 0,4 \times CD$

AVALIAÇÃO

Datas Importantes

- 2ª e 3ª Semana (17/Fev-28/Fev)
Definição dos grupos
- 3ª e 4ª Semana (24/Fev-6/Mar)
Escolha dos temas de trabalho
- 9ª Semana (10/Abr, 23:59)
Entrega 1ª Parte (via Moodle) + Apresentação
- 15ª Semana (22/Maio, 23:59)
Entrega 2ª Parte (via Moodle + GitHub) + Apresentação

AVALIAÇÃO

Atividades extras com possibilidade e bónus

- *Kahoot!*
Questões teóricas sobre a matéria lecionada nas aulas teóricas
- *HackerRank*
Exercícios diversos, a serem realizados em formato de competição de programação

GESTÃO DA UC

Informação da Unidade Curricular

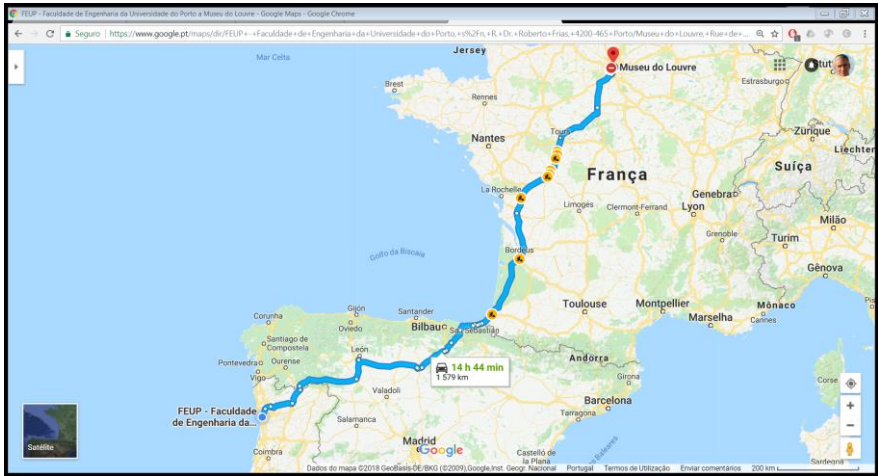
- [SiFEUP](#)

Planeamento, datas e material de apoio

- Moodle

EXEMPLOS DE PROBLEMAS: CAMINHO MAIS ‘CURTO’

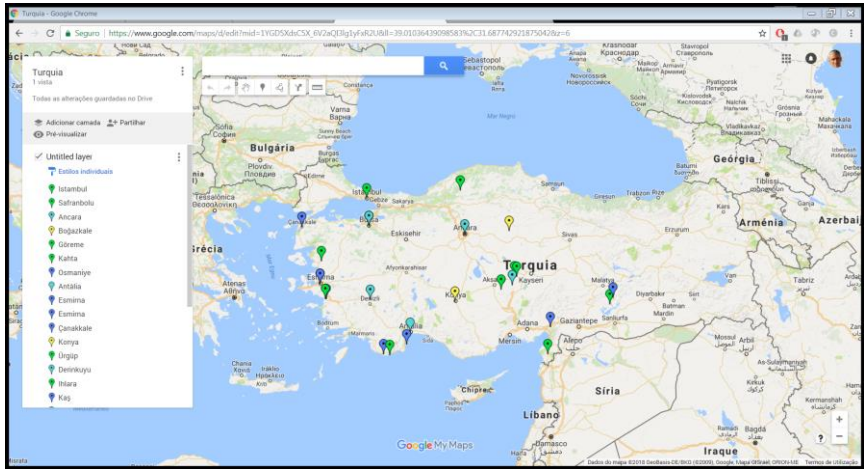
Qual o caminho mais curto / mais rápido / mais barato entre 2 pontos?



Abstraido como problema em grafos, resolúvel em tempo linear.

EXEMPLOS DE PROBLEMAS: PROBLEMA DO CAIXEIRO VIAJANTE

Qual o melhor circuito para passar nos pontos de interesse assinalados?



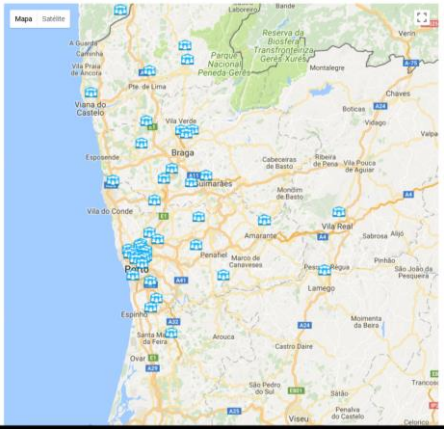
Abstraido como problema em grafos, em geral não resolúvel em tempo linear.

EXEMPLOS DE PROBLEMAS: PROBLEMAS DE EMPARELHAMENTO

Todos os anos, dezenas de milhares de professores candidatam-se a vagas nas escolas.

Existem preferências de parte a parte (escolas e professores).

Interessa maximizar o nº de vagas ocupadas.



Abstraído como problema de emparelhamento em grafos bipartidos (mais precisamente, problema dos casamentos estáveis), tratável em tempo polinomial.

<http://www.dcc.fc.up.pt/Pubs/TR05/dcc-2005-02.pdf>