

# CAL

## Concepção e Análise de Algoritmos

2015-2016 (2S)

Rosaldo Rossetti & Ana Paula Rocha

## Docentes



Rosaldo Rossetti  
([rossetti@fe.up.pt](mailto:rossetti@fe.up.pt))



Ana Paula Rocha  
([arocha@fe.up.pt](mailto:arocha@fe.up.pt))

## Objectivos

- Gerais
  - complementar e aprofundar as disciplinas de “Programação” e de “Algoritmos e Estruturas de Dados”
  - Introduzir técnicas de concepção e implementação de algoritmos eficientes p/ 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 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 p/ 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
- **Representação e formalização** de algoritmos, **análise** da sua complexidade (temporal e espacial); **verificação** e **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
  - M.A. Weiss. **Data Structures and Algorithm Analysis in C++**, 3/E. New York, NY: Addison Wesley, 2007.
  - T. Cormen; C. Leiserson; R. Rivest; C. Stein. **Introduction to Algorithms**. Cambridge, MA: MIT Press, 2009.
  - 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: Grafo (50%)  $\geq 8.0$  valores
    - Parte II: Grafo, Strings/Texto/Ficheiros (50%)  $\geq 8.0$  valores
  - Nota mínima em cada trabalho: 40%
- **Exame final (EF)  $\geq 8.0$  valores**
  - Consulta condicionada; duração 2 horas
- **Avaliação Final**
  - $AF = 0,6 \times EF + 0,4 \times CD$

## Avaliação

- **Datas Importantes**
  - 2ª Semana (22/Fev-26/Fev)  
Definição dos grupos
  - 3ª Semana (29/Fev-04/Mar)  
Escolha dos temas de trabalho
  - 8ª Semana (11/Abril, 12:00)  
Entrega da Parte I (via Moodle) + Apresentação
  - 14ª Semana (30/Maio, 12:00)  
Entrega Parte II (via Moodle) + Apresentação

## Gestão da UC

- Informação da Unidade Curricular
  - [SiFEUP](#)
- Planeamento, datas e material de apoio
  - Moodle
  - TBD...