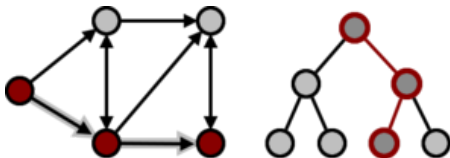


# Desenho e Análise de Algoritmos

Pedro Ribeiro

DCC/FCUP

2020/2021



- Site: <http://www.dcc.fc.up.pt/~pribeiro/aulas/daa2021/>
- Recursos Principais:
  - ▶ **Piazza:** Fórum para perguntas e dúvidas gerais
  - ▶ **Mooshak:** Submissão de código para problemas e resposta a quizzes
  - ▶ **YouTube:** Vídeos (pré-gravados) das aulas teóricas
  - ▶ **Slack:** Esclarecimento de dúvidas em formato "*instant messaging*"

# Obtenção de Frequência

- Não serão registadas presenças (teóricas e práticas)
- Semanalmente, serão feitos **questionários**:
  - ▶ São obrigatórios, mas não contam para nota
  - ▶ Cada um estará online durante uma semana (00:01 de Domingo a 23:50 de Sábado)  
(depois ficam disponíveis para treino, sem contar para frequência)
  - ▶ São constituídos por perguntas de escolha múltipla
  - ▶ Podem ver os resultados, saber que opção estava correcta e voltar a submeter quantas vezes quiserem
  - ▶ Para obter frequência é necessário ter respondido a 50%  
**11 questionários → têm de responder a 6**

# Fórmula de Cálculo da Avaliação

- **P:** nota prática, valendo **30%** da nota final, obtida através de 3 componentes:
  - ▶ 2 testes práticos de programação (2.5 valores cada)
  - ▶ resolução de exercícios ao longo do semestre (1 valor).

Nota mínima:  $P \geq 1.5$  (escala da nota: 0 a 6).

- **EN:** nota do exame de época normal, valendo 70% da nota final, obtida através de um exame escrito (presencial) com nota de 0 a 20
- **ER:** na época de recurso será feito um único exame (presencial), valendo 70% da nota final, não sendo possível repetir a componente prática

**Classificação da época normal:**  $C = EN * 0.7 + P \geq 9.5$

**Classificação da época de recurso:**  $C = ER * 0.7 + P \geq 9.5$

# Sobre a componente prática

- Poderão usar C, C++ ou Java
- Resolução de exercícios ao longo do semestre
  - ▶ 12 aulas com exercícios "pontuáveis"
  - ▶ Cada aula vale 10% (máximo=100%)
  - ▶ Os exercícios estarão disponíveis durante 3 semanas (depois ficam disponíveis para treino, sem contar para avaliação)
  - ▶ Devem reportar no código qualquer ajuda que tenham recebido
- Testes Práticos
  - ▶ Serão de no máximo 2 horas
  - ▶ Terão objectivos específicos divulgados
  - ▶ Terão acesso a código vosso submetido antes do teste (compensa por isso terem feito os exercícios antes, perceberem bem o que fizeram e terem código organizado)

# Datas dos Testes

*Estas datas são neste momento provisórias:*

- **1º teste prático de programação: 31 de Outubro**
- **2º teste prático de programação: 12 de Dezembro**

# Pré-requisitos

- Conhecimentos de **C/C++** ou **Java**
- Conhecimentos de **algoritmos básicos** (contagem, pesquisa, ordenação, ...)
- Conhecimentos de **estruturas de dados básicas** (arrays, listas, pilhas, filas, ...)
- Preferencialmente ter concluído as unidades curriculares de **"Programação Imperativa"** e **"Estruturas de Dados"** (ou equivalente)

# Objetivos da Unidade Curricular

Competência na área de técnicas de **concepção e análise de algoritmos eficientes**:

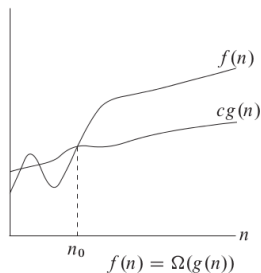
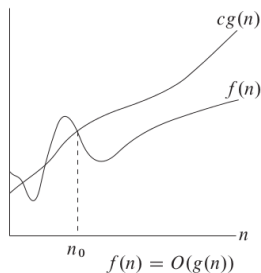
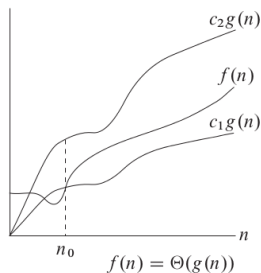
- Competência na **análise da complexidade de algoritmos** e compreensão de algumas classes de complexidade
- Enriquecimento do conhecimento sobre **modelos genéricos de tipos de problemas e técnicas algorítmicas** a eles associadas.
- **Experiência prática** na aplicação de algoritmos genéricos a problemas concretos.



# Visão Geral do Programa

## Análise assintótica do tempo de execução de algoritmos:

- Notação *Big O* ( $O$ ,  $\Omega$  e  $\Theta$ )
- Análise de programas iterativos e recursivos
- Previsão de tempo de execução

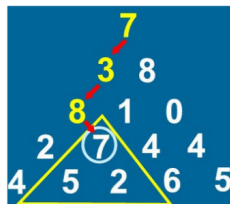
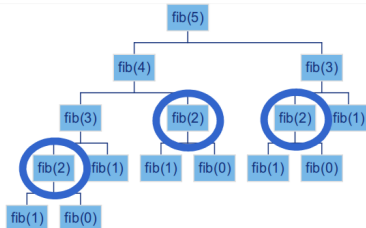


# Visão Geral do Programa

## Técnicas de Desenho de Algoritmos

- Pesquisa exaustiva (*Força Bruta*)
- Dividir para conquistar
- Algoritmos *greedy*
- Programação dinâmica

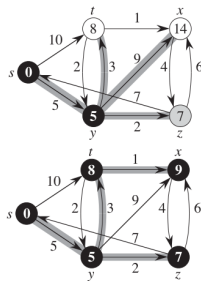
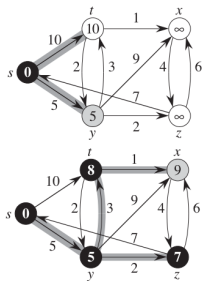
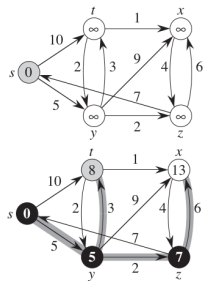
i \ j	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	1	2	3	3	4
2	2	2	2	2	3	4
3	3	3	3	3	3	4
4	4	3	4	4	4	3
5	5	4	4	5	5	4



# Visão Geral do Programa

## Algoritmos de grafos

- Representação de grafos
- Pesquisa em largura e pesquisa em profundidade
- Árvores de cobertura mínima
- Caminhos mínimos
- Redes de fluxo



# Visão Geral do Programa

## Algumas estruturas de dados especializadas

- Filas de prioridade
- Conjuntos disjuntos
- Árvores binárias de pesquisa equilibradas

