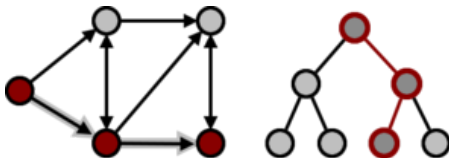


Desenho e Análise de Algoritmos

Pedro Ribeiro

DCC/FCUP

2016/2017



Informações Gerais

- Site: <http://www.dcc.fc.up.pt/~pribeiro/aulas/daa1617/>
- Piazza: <http://piazza.com/up.pt/fall2016/cc2001>
(usado para toda a comunicação nesta UC)
- Aulas **teóricas**:
 - ▶ 3ª Feira: 15:00 às 16:00 (Sala 0.41, FC4)
 - ▶ 6ª Feira: 15:00 às 16:00 (Sala 0.41, FC4)
- Aulas **práticas**:
 - ▶ PL1: 4ª Feira: 14:00 às 16:00 (Lab8, Luisa Lima)
 - ▶ PL2: 3ª Feira: 16:00 às 18:00 (Lab2, Pedro Ribeiro)
 - ▶ PL3: 6ª Feira: 16:00 às 18:00 (Lab2, Pedro Ribeiro)
 - ▶ PL4: 4ª Feira: 16:30 às 18:30 (Lab8, Luisa Lima)
 - ▶ PL5: 2ª Feira: 16:00 às 18:00 (Lab3, Pedro Ribeiro)
- **Atendimento** (gabinete 1.03 - irei mudar para o 1.47):
 - ▶ 3ª feira: 11:00 às 12:30 (almoço DAA - avisar e/ou aparecer às 12:30)

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%
12 questionários → têm de responder a 6

Fórmula de Cálculo da Avaliação

- **P:** nota prática, valendo **35%** da nota final, obtida através de 3 componentes:
 - ▶ 2 testes práticos de programação (2.5 valores cada)
 - ▶ resolução de exercícios ao longo do semestre (2 valores).

Nota mínima: $P > 2$ (escala da nota: 0 a 7).

- **T:** nota teórica, valendo **65%** da nota final, obtida através da média de dois testes escritos (TE1+TE2), com nota de 0 a 20 (TE1 e TE2 incidirão sobre conjuntos de matérias diferentes). Nota mínima em cada teste: 5 (de 0 a 20).
- **R:** na época de recurso será feito um único exame, com nota de 0 a 20, com a matéria correspondente aos dois testes escritos. Nota mínima: 5.

Classificação da época normal: $C = T * 0.65 + NP \geq 9.5$

Classificação da época de recurso: $C = R * 0.65 + NP \geq 9.5$

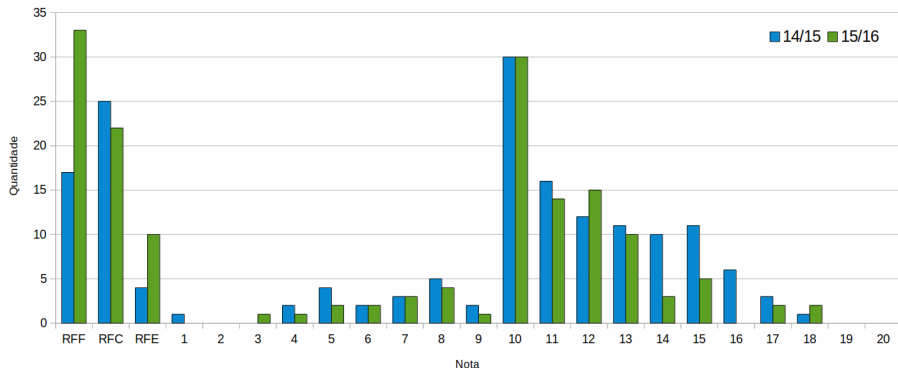
Sobre a componente prática

- Poderão usar C, C++ ou Java
- Resolução de exercícios ao longo do semestre
 - ▶ 11 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 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)

Estas datas são neste momento provisórias:

- **1º teste escrito:** Início de Novembro
- **2º teste escrito:** Janeiro (época normal de exames)
- **1º teste prático de programação:** Novembro
- **2º teste prático de programação:** Início de Janeiro

Distribuição das notas de DAA



- Total de Alunos: **165** (14/15), **160** (15/16), **159** (16/17)
- Alunos aprovados
 - ▶ % total inscritos: **60.6%** (14/15), **50.6%** (15/16)
 - ▶ % total com frequência: **67.6%** (14/15), **63.8%** (15/16)
 - ▶ % total com nota mínima e exame: **84.0%** (14/15), **85.3%** (15/16)

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 **"Introdução à Programação"** e **"Estruturas de Dados"**
(ou equivalente)

Objectivos da Unidade Curricular

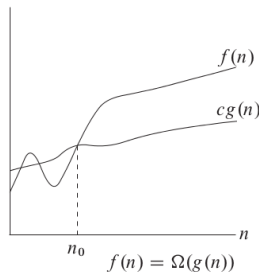
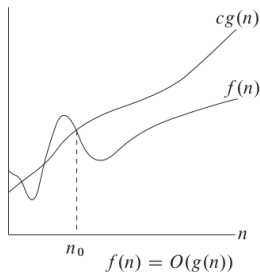
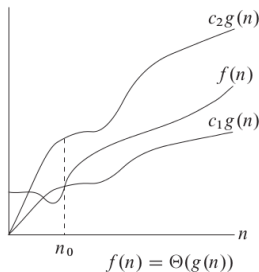
Competência na area de técnicas de **concepção e análise de algoritmos eficientes**:

- 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 , Ω e Θ)
- Análise de programas iterativos e recursivos
- Previsão de tempo de execução

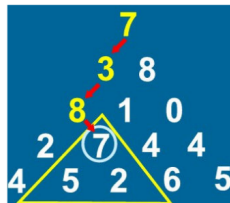
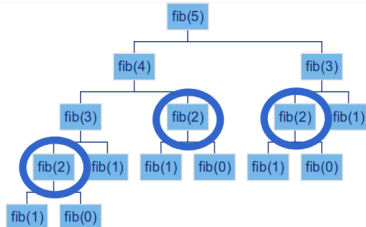


Visão Geral do Programa

Técnicas de Desenho de Algoritmos

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

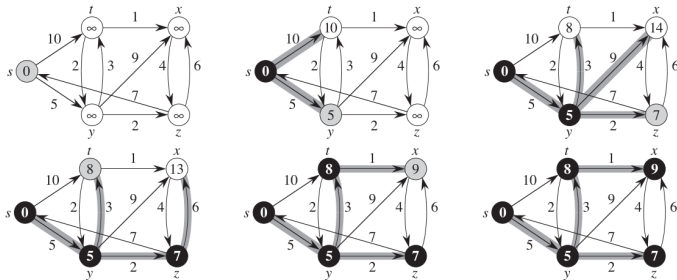
i \ j	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	1	2	3	3	4
2	2	2	2	2	3	4
3	3	3	3	3	3	4
4	4	3	4	4	4	3
5	5	4	4	5	5	4



Visão Geral do Programa

Algoritmos de grafos

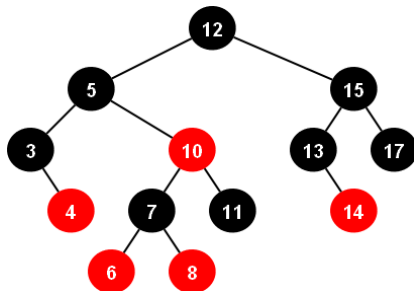
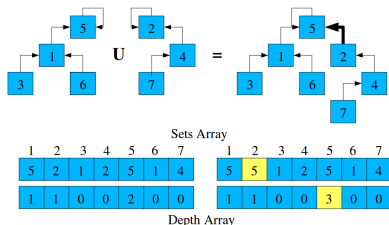
- Representação de grafos
- Pesquisa em largura e pesquisa em profundidade
- Árvores de cobertura mínima
- Caminhos mínimos
- Redes de fluxo



Visão Geral do Programa

Algumas estruturas de dados especializadas

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



Funcionamento das aulas

- **Teóricas:** slides +
uso do computador do docente +
exposição no quadro
- **Práticas:** guião com exercícios (em "papel") +
implementação em código (C/C++ ou Java)
- **Material auxiliar:** divulgado no site, em português e/ou inglês
slides, apontamentos, animações, applets, ...