

INTRODUÇÃO A TÉCNICAS ALGORÍTMICAS E COMPUTABILIDADE

Ana Neri
ana.i.neri@inesctec.pt

DI,
Universidade do Minho

13 Fevereiro 2023

PARTE I: CONTEÚDO

- 1 Introdução à cadeira: uma visão global da matéria 4**
 - 1.1 Importância dos algoritmos e da sua análise 5
 - 1.2 Complexidade 7
 - 1.3 Estruturas discretas e algoritmos associados 8
 - 1.4 Algoritmos sobre grafos 9

PARTE II: CONTEÚDO

1	Estrutura do Curso e Pragmatismos	12
1.1	Objetivos e Resultados de aprendizagem	12
1.2	Programa	14
1.3	Avaliação	15
1.4	Momento de avaliação	16
1.5	Outras informações	17

Part I

TÉCNICAS ALGORÍTMICAS E COMPUTABILIDADE

TÉCNICAS ALGORÍTMICAS E COMPUTABILIDADE

O que são algoritmos?

Um algoritmo é um procedimento computacional bem definido que aceita um valor (ou conjunto de valores) como input e produz um valor (ou conjunto de valores) como output.

O que são técnicas algorítmica?

Técnicas algorítmicas são métodos específicos aplicados a algoritmos para torná-los mais eficientes ou fáceis de implementar.

O que é computabilidade?

A computabilidade é a área que se concentra no estudo da capacidade e limitações da computação, abrangendo a identificação de algoritmos que possam ser executados por máquinas de computação e o estudo dos limites da solução de problemas através de algoritmos.

IMPORTÂNCIA DOS ALGORITMOS

Alguns exemplos de algoritmos na indústria:

- ▶ Otimização de rotas
- ▶ Recomendação
- ▶ Análise financeira
- ▶ Detecção de fraude
- ▶ ...

ALGORITMOS E EFICIÊNCIA

Quando procuramos um algoritmo interessa-nos que este consiga ser **eficiente**.

Há várias formas de avaliar a eficiência de um algoritmo, uma delas é tempo de execução.

Dentro dessa análise, um problema para o qual não se conhece uma solução eficiente classifica-se como **NP**.

Se um problema é **NP-completo**, é uma subclasse dos anteriores que

- ▶ são aparentemente simples;
- ▶ não se conhece um algoritmo que os resolva de forma eficiente;
- ▶ têm várias aplicações;
- ▶ se uma solução for encontrada para um deles, pode ser usada para encontrar uma solução para todos os outros problemas NP-completo
- ▶ Em geral é necessário encontrar soluções aproximadas ou aplicar técnicas heurísticas para tentar resolver esses problemas.

COMPLEXIDADE

Para além de queremos garantir que o programa está correto, também queremos saber que recursos são exigidos.

Alguns exemplos destes recursos são **espaço, tempo e energia**.

Como definem complexidade de espaço?

É o espaço de memória exigido em função do tamanho (n) da entrada.

e como definem complexidade de tempo?

É o tempo exigido em função do tamanho (n) da entrada.

A complexidade pode ser estimada para:

- ▶ o “melhor caso”
- ▶ o “pior caso”
- ▶ o “caso médio”

ESTRUTURAS DISCRETAS

Escolher estruturas de dados erradas pode atrasar o vosso programa ou pior. Por isso, nesta aula vamos explorar em detalhe estruturas discretas e os seus algoritmos associados.

O que são estruturas discretas?

Podemos definir uma estrutura discreta como uma estrutura na qual os objetos têm valores distintamente separáveis em vez de contínuos.

Alguns exemplos de estruturas discretas?

- ▶ Sequências
- ▶ Listas ligadas
- ▶ Pilhas
- ▶ Árvores
- ▶ Grafos
- ▶ Automatos
- ▶ ...

Vamos usar estas estruturas para analisar **algoritmos de Árvores e Grafos**

ALGORITMOS SOBRE GRAFOS

Dentro de algoritmos de grafos está previsto explorarmos estratégias “greedy” e programação dinâmica.

O que é uma estratégia “greedy”?

Um algoritmo que utilize estratégias “greedy” é um que em cada estágio escolhe uma solução localmente ótima, esperando, no fim, obter uma solução global ótima. São tipicamente utilizados para problemas de otimização.

O que é programação dinâmica?

A programação dinâmica é um método de construção de algoritmos que otimizam a resolução do problemas através de análise combinatória de subproblemas, de forma evitar a resolução de problemas idênticos múltiplas vezes.

EM SUMA:

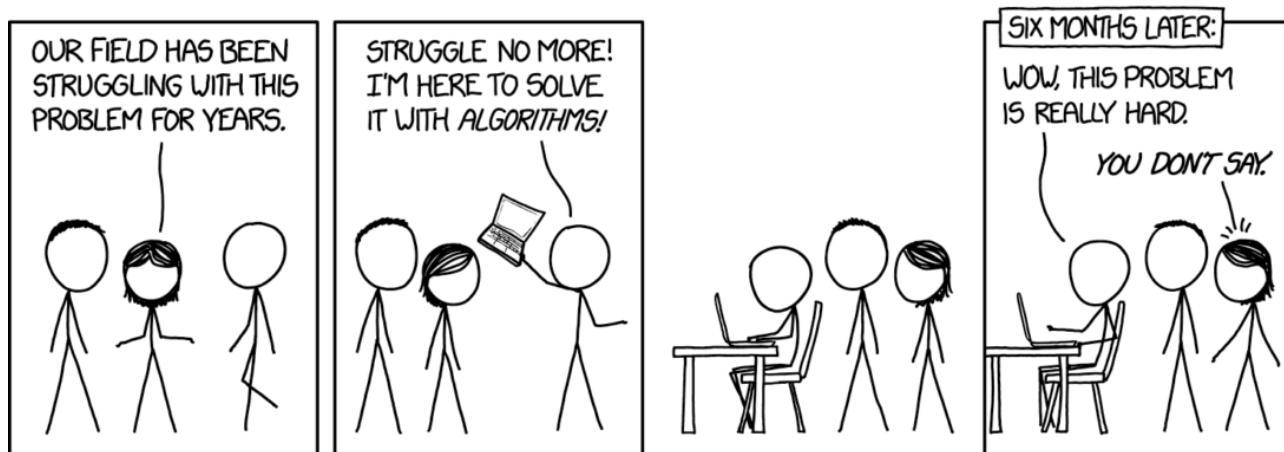


Figure. Ter algoritmos para resolver o problema pode não ser suficiente, nesta aula vamos analisar isto em mais detalhe.

Part II

ESTRUTURA DO CURSO E PRAGMATISMOS

OBJETIVOS

Esta unidade curricular usa paradigma imperativo de programação para introduzir:

- ▶ técnicas para o desenho de algoritmos (seguindo estratégias algorítmicas comuns e utilizando estruturas de dados clássicas), e para a análise rigorosa da sua eficiência.
- ▶ e noções de computabilidade e estruturas discretas para modelação de dados, nomeadamente grafos e estruturas ordenadas.

RESULTADOS DE APRENDIZAGEM

- ▶ Dividir um problema em subproblemas mais simples;
- ▶ Definir tipos de dados para a representação dos dados e resultados de um dado problema;
- ▶ Analisar e desenvolver algoritmos manipulando diversos tipos de estruturas discretas, nomeadamente, ordens, árvores e grafos;
- ▶ Determinar a complexidade assintótica de um algoritmo iterativo ou recursivo;
- ▶ Pressupões reconhecer e utilizar estratégias algorítmicas fundamentais;
- ▶ Aplicar a noção de computabilidade (máquina de Turing, função computável) e problemas associados;
- ▶ Resolver problemas NP-completos e alguns algoritmos aproximados para a sua resolução.

PROGRAMA

Parte I:

- ▶ Análise de complexidade
- ▶ Classes de problemas

Parte II:

- ▶ Estruturas discretas
- ▶ Algoritmos sobre grafos
- ▶ Programação dinâmica

AVALIAÇÃO

- ▶ 2 teste para avaliar as aulas T;
- ▶ e 1 trabalho prático para avaliar a TP.

Avaliação da componente teórica resulta da média aritmética dos dois testes (cada um com nota mínima de 7 valores), ou do exame da época de recurso, este não contabiliza a nota obtida nos testes e aborda toda a matéria lecionada durante o semestre.

A componente teórico-prática é avaliada através de um relatório e de uma apresentação oral do trabalho feito.

MOMENTOS DE AVALIAÇÃO

Testes:

Teste 1	20 de março de 2023
Teste 2	15 de maio de 2023

Projeto:

Entrega de relatório	5 de junho 2023 (16h00)
Defesa	7 de junho de 2023 (manhã)

OUTRAS INFORMAÇÕES

Horário para dúvidas:

- ▶ Presencial: segundas 9h-10h
- ▶ Online: quartas 11h-13h

Anúncios:

- ▶ Blackboard
- ▶ Website (<http://lmf.di.uminho.pt/tecnicas-algoritmicas-computabilidade-2223/>)
- ▶ Slack (Enviem-me um email com nome e n° de aluno para vos mandar o convite)