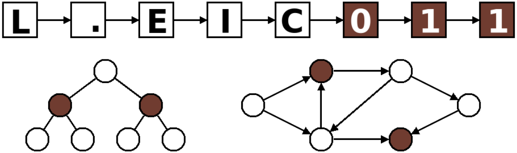
Introduction

L.EIC

Algoritmos e Estruturas de Dados

2025/2026



P Ribeiro, AP Tomás

General Information

- Main Resources:
  -
    Algorithms and Data Structures
    2025/2026 Edition
    Bachelor in Informatics and Computing Engineering
    L . E I C 0 1 1

    Webpage: course webpage (classes, slides and all relevant information)
    https://www.dcc.fc.up.pt/~pribeiro/aulas/aed2526/
  - **Discord:** communication/discussion (*"instant messaging"* format)
    (invite already sent by email)
  - **Mooshak:** Code submissions with automatic test evaluation
    https://mooshak.dcc.fc.up.pt/~aed/
    (passwords will be sent by email)

Teaching Staff

- **Theoretical Classes** *(one 2h class per week)*
  - Ana Paula Tomás (FCUP) - *1st Half of Semester*
  - Pedro Ribeiro (FCUP) - *2nd Half of Semester*

- **Practical Classes:** *(one 2h class per week)*
  - Ana Paula Tomás (FCUP)
  - Bernardo Leite (FEUP)
  - Filipa Ramos (FEUP)
  - Iohan Soares (FEUP)
  - Jadna Cruz (FEUP)
  - Jorge Oliveira (FEUP)
  - Tiago Carvalho (FEUP)
  - Vanessa Silva (FCUP)
  - Vasco Cruz (FCUP)

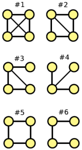
Teaching Staff - Ana Paula Tomás

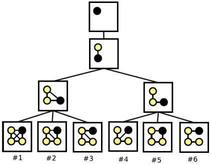
Name: Ana Paula Tomás
Office: FC6 1.15 (at DCC/FCUP - Campo Alegre)
Website: `https://www.dcc.fc.up.pt/~apt/`
Course Director of L:CC

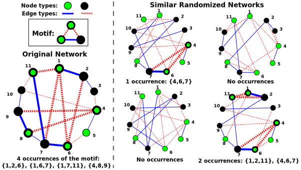
Main Research Interests

- Design and Analysis of Algorithms
- Geometric and Network Problems
- Discrete Optimization and Constraint Programming
- Enumerative Combinatorics
- Computer Assisted Learning

Teaching Staff - Pedro Ribeiro







Name: Pedro Ribeiro
Office: FC6 1.47 (DCC/FCUP)
Website: `https://www.dcc.fc.up.pt/~pribeiro/`
Course Director of M:CC

Main Research Interests

- Complex Networks, Network Science, Graph Mining, Data Science
- Algorithms and Data Structures, Complexity
- Parallel and Distributed Computing
- Computer Science Education and Programming Contests

Pedro Ribeiro - Competitive Programming

I am involved in many **algorithmic programming contests**
(organization, creating and solving problems, student mentoring and training, ...)

- Pre-University Students (Primary and Secondary Education)
  - **Olympiads** in Informatics (National and International, e.g., ONI, IOI)
  - **Bebras** - International Computational Thinking Challenge
- University Students
  - National and International **ICPC** Contests (e.g., MIUP, SWERC, EUC)

Pre-requirements

- Students should have basic knowledge of programming and **C++**

```cpp
// Your first C++ Program

#include <iostream>

int main() {
  std::cout << "Hello World!" << std::endl;
  return 0;
}
```

Hopefully not really your FIRST program 🙂

Methodology

Theoretical classes

- on site (”*face to face*” classes)
- (formal) exposition of the subjects, presentation of examples, analysis and discussion, some livecoding

Practical classes

- programming exercises in **C++** *(we will use C++17)*
- no imposed IDE (suggestion: VSCode, CLion)
- automatic evaluation (`Mooshak` and test cases)

Evaluation

Final Mark $= 0.7 \times Exam + NP \geq 9.5$

- *NP*: 2 practical tests with automatic evaluation (2.5 points each) and exercises during the semester (1 point)
- In *"Recurso"*, only the Exam component can be improved

Frequency

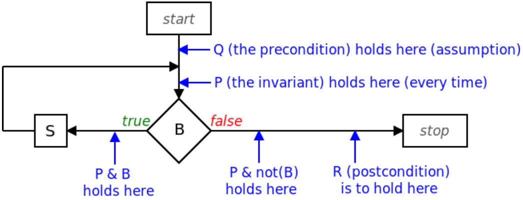
Students may not exceed the limit of absences (**25%** of TP classes)

Objectives

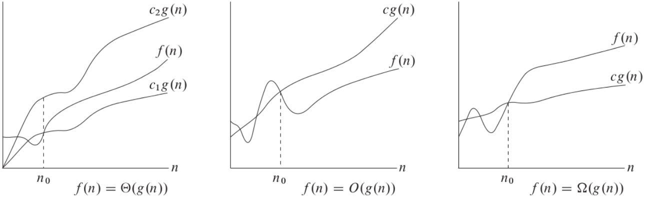
- Analyze the **correctness of simple algorithms**
- Analyze the **temporal and spatial complexity of algorithms**
- Understand the concept of **abstract data type** and know how to organize programs around this concept
- Know the **fundamental data structures and associated algorithms** and respective complexity
- **Choose appropriate** collections, data structures and algorithms to solve practical problems
- **write programs in C++** that implement and use the fundamental data structures and algorithms

An overview of the program

- Algorithmic **correctness** (and the concept of **loop invariant**)

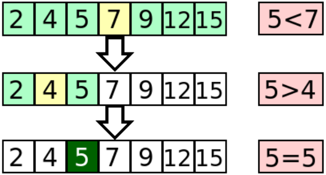


- Algorithmic **efficiency** (time and space)
  (asymptotically analysis and the **Big O notation**)

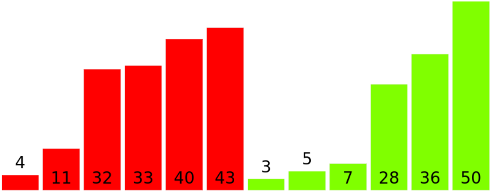


An overview of the program

- **Searching** algorithms (linear search, binary search and variants)

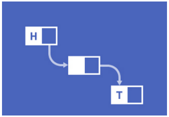


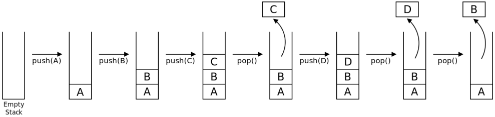
- **Sorting** algorithms (comparison based sorting and linear sorting) *[e.g. MergeSort, QuickSort, RadixSort, CountingSort, ...]*



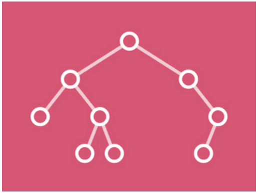
An overview of the program

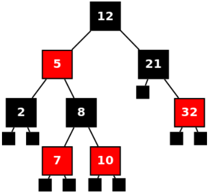
- Linear data structures: **lists**, **stacks**, **queues**





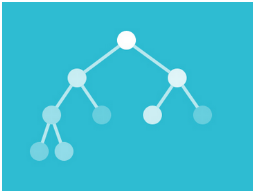
- Hierarchical data structures: **binary trees**, **binary search trees**, **balanced binary trees** and variants *(e.g. AVL and Red-Black trees)*

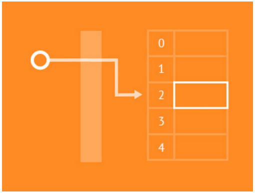




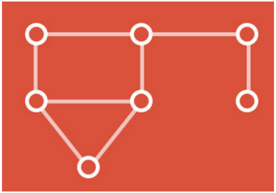
An overview of the program

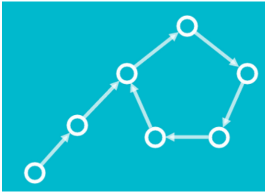
- Other essential data structures (**priority queues**, **hash tables**)





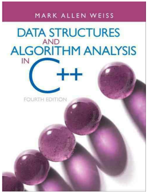
- **Graphs** (concept, representation, traversal and fundamental algorithms) *(e.g. DFS, BFS and applications)*



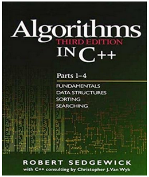


Bibliography - Main Books

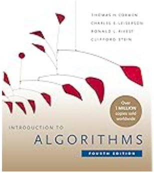
- **Data Structures & Algorithm Analysis in C++**
  Mark Allen Weiss, 4th Edition, Pearson Education



- **Algorithms in C++**
  Robert Sedgewick, 3rd Edition, Princeton University

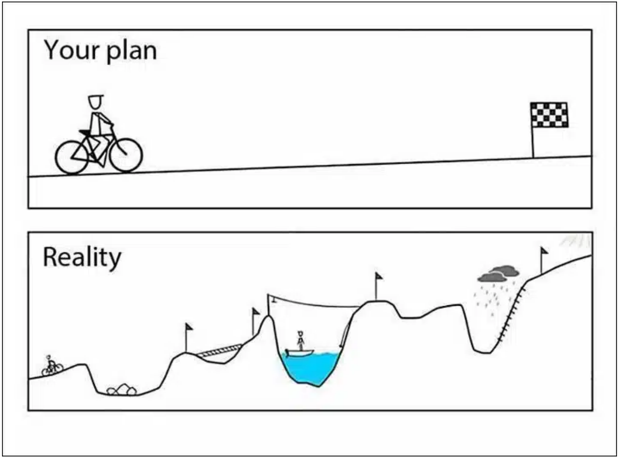


- **Introduction to Algorithms**
  TH Cormen, CE Leiserson, RL Rivest and C Stein
  4th Edition, MIT Press (also known as **CLRS**)



Good Work!

`Genius is 1% inspiration and 99% perspiration` (Thomas Edison)



Source: Mark Rober's **Super Mario Effect** Tedx Talk
`https://youtu.be/9vJRopau0g0`