

Algoritmos e Estruturas de Dados

Licenciatura em Engenharia Informática e Computação
2021/2022

AP Rocha, P Ribeiro,
R Rossetti, F Ramos, L Grácio, A Costa, S Martins

- Course moodle page
- Slack: <https://aed2122.slack.com>
used for communication/discussion (*registration link available on moodle*)
- Theoretical classes:
Ana Paula Rocha
Pedro Ribeiro
- Practical classes:

Ana Paula Rocha

Pedro Ribeiro

António Costa

Rosaldo Rossetti

Filipa Ramos

Sofia Martins

Luciano Grácio

- Theoretical classes
 - online using zoom
 - formal exposition of the subjects, presentation of examples, analysis and discussion.
- Practical classes
 - programming exercises in C++ using unit tests: CLion, Google Tests
 - optional code submission to an automatic judging platform (Mooshak)
 - group project, but most will be outside of classes

Final Mark = $0.3 \cdot \text{CIP} + 0.3 \cdot \text{CIT} + 0.4 \cdot \text{CG}$

- A minimum mark of 40% is required in every assessment component (CIP, CIT, CG1, CG2)

- **Individual Component – 60%**

- *Practical Component (CIP)* - 30%
 - practical on computer evaluation, programming assignments using unit tests: CLion, Google Tests
- *Theoretical Component (CIT)* - 30%
 - multiple-choice questions

- **Group Component (CG) - 40%**

- two small projects (CG1 and CG2) to be implemented in group (3 students)

Evaluation

- There will be a recovery test
 - for students that did not obtain the minimum required in the component CIP or CIT.
 - The grade of this test is limited to 50%.
- Students registered under any special status:
 - group assessment may be performed individually
 - student must talk to the lecturer to make all the arrangements and fix a reasonable schedule.
 - should attend and perform the individual assessment components, as normally scheduled
- Student may not exceed the limit of absences (25% of classes)

Pre-requirements

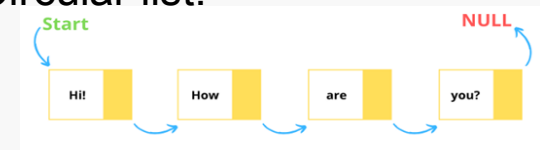
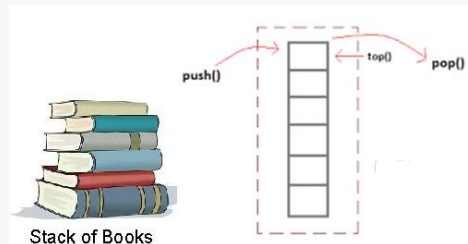
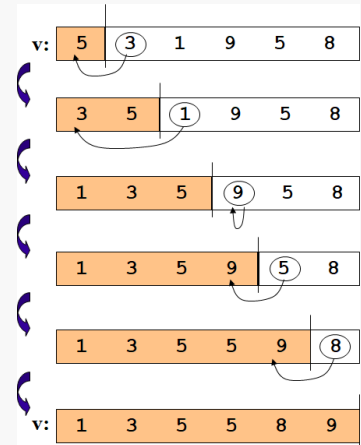
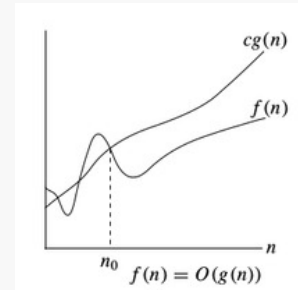
- Basic knowledge of programming and C++

Objectives

- Analyze the **correctness of simple algorithms** (and test experimentally)
- Analyze the **temporal and spatial complexity** of algorithms (and measure experimentally)
- 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

Program

- Analysis of algorithm correctness
- Algorithmic efficiency
 - temporal and space complexity
- Searching and sorting algorithms
 - Sequential search, binary search
 - InsertionSort, BubbleSort, ShellSort, MergeSort, QuickSort, CountingSort.
- Linear data structures
 - Lists
 - Array-based and linked list. Doubly linked lists. Circular list.
 - Stacks
 - Queues

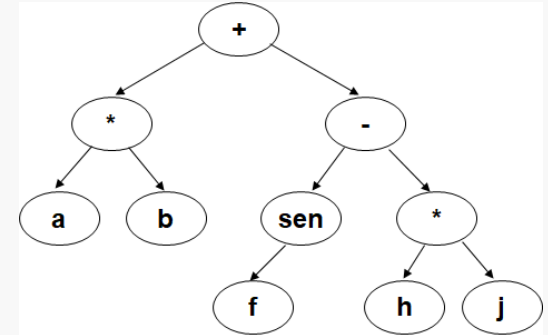
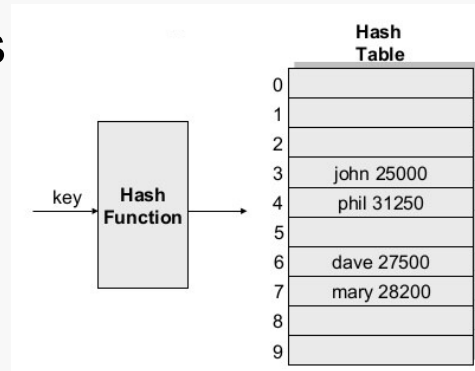


Program

- Hierarchical data structures
 - Binary Trees, Binary Search Trees, Balanced binary trees and variants

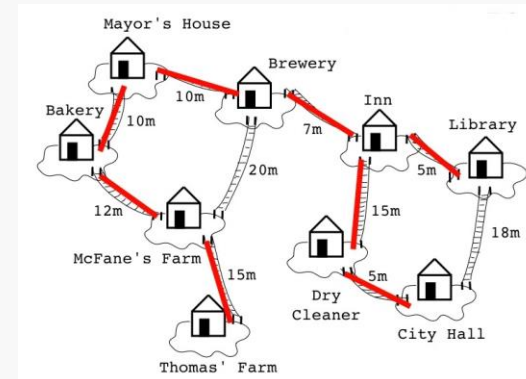
- Other data structures

- Hash Tables
- Priority Queues
- Disjoint-sets

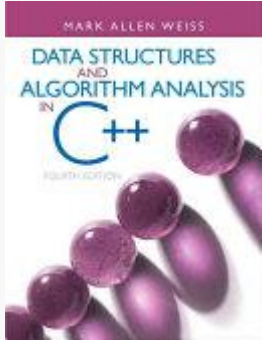


- Graph algorithms

- Representation
- DFS, BFS and applications
- Minimum Spanning Trees
 - Prim, Kruskal
- Shortest Paths
 - Dijkstra and variants, Bellman-Ford, Floyd-Warshall



Bibliography



Data Structures & Algorithm Analysis in C++
Mark Allen Weiss, 4th Edition, Pearson Education,
2014

additional:

- Algorithms in C++
Robert Sedgewick, 3rd Edition, Princeton University, 2002