Algorithmic Strategies 2025/26

Week 1 – Introduction

UNIVERSIDADE D
COIMBRA

Outline

Luís Paquete

- T1 + PL3 + PL4 + PL5
- paquete@dei.uc.pt
- D.2.18 by e-mail
- Wed, 9h-14h00

Noé Godinho

- PL1 + PL2
- noe@dei.uc.pt
- D.3.13 by e-mail
- Mon, 14h-16, 18h-19h
- Wed, 15h-16, 18h-19h

Algorithmic Strategies: what is it about?

- Course about programming and algorithmic thinking
- Focused on problem solving through the study of algorithm paradigms and their application to problems in different domains: Optimization, Graphs, Geometry, Strings, ...
- Re-uses programming techniques that students already know

Introduction

Knowledge:

- Algorithmic paradigms (backtracking, DP, greedy, etc.)
- Problem decomposition
- Structural properties of problems
- Methods for algorithm correctness
- Graph algorithms

Skills:

- Model a real-world problem as a computational problem
- Identify suitable problem decompositions
- Select an appropriate algorithmic paradigm for a problem
- Prove algorithm correctness and derive test (edge) cases
- Derive time and space complexity
- Implement correct and efficient programs
- Experimentally evaluate performance

How it works?

- T class: Introduction to the topic and to the programming problem.
- PL class: Support on programming and written exercises, clarification of doubts and individual training.
- Assignments: programming problems and written exams.

Assessment

A) 3.5 points in two programming problems

B) 0.5 points in PL programming exercises

C) 16.0 points in one written exam.

Minimum requirements: 35% in C)

Retake and special exam only for C)

Assessment

A - Programming problems

- Two problems to be solved in teams of two with Mooshak, partial points are allowed.
- Test cases have different levels of difficulty.
- Students may need to orally defend the grade obtained on both programming problems.
- Register at mooshak with team name

  <student 1 ID>-<student 2 ID>

  Nonconforming registrations will be deleted from mooshak.

Assessment

B - PL Programming exercises

- Maximum 10 problems to be solved individually with Mooshak, 0.05 points (accepted in mooshak) each. Consider this to improve your coding skills.
- A problem is released every week on Tuesdays and stays open for two weeks. The solution to the problem is discussed in the PL classes in the first week before the release.
- Register at mooshak (EA2026_PL) with team name <student ID>. Nonconforming registrations will be deleted from mooshak.

Assessment

C - Written exam

- Totals 16 in the official scale.
- One exam to be solved individually
- Closed books
- It tests your knowledge!

Calendar

	T1 day	**Topic**
1	18 Feb	Intro and problem solving
2	20 Feb	Recursion
3	27 Feb	Backtracking & Problem 1 (27/2)
4	6 Mar	Dynamic programming
5	13 Mar	Dynamic programming
6	20 Mar	Greedy algorithms
7	27 Mar	Branch and bound & 1st Deadline (27/3)
8	10 Apr	Graph algorithms
9	17 Apr	Graph algorithms & Problem 2 (17/4)
10	24 Apr	Graph algorithms
11	1 May	Graph algorithms (no T1, only PLs)
11	8 May	Graph algorithms
12	15 May	Computational geometry & 2nd Deadline (21/5)

Deadlines for problems are at 23h59m

Assessment

- Teams: Students **must** split into groups of two. Students cannot change group after team assessment begins. Meet your team member at least once a week to solve problems.
- Programming languages: Use `C`, `C++`, `Java` and `Python`. However, some large test cases may not be suitable for `Java` and `Python`.
- Use PLs and teacher attendance timetable for clarification of doubts.
- Mooshak automatically opens and closes contests. No delays are tolerated!

Bibliography

Main references

- Algorithms, Jeff Erickson (draft available here)
- Introduction to Algorithms, T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein

Additional references

- Programming Challenges, S. Skiena, M. Revilla
- Algorithm Design, J. Kleinberg, E. Tardos
- The Algorithm Design Manual, S. Skiena (2nd edition)
- How to Think About Algorithms, J. Edmonds
- The Art of Computer Programming, D. Knuth
- C++ STL and JDK