

# Programming Research Algorithms

School of Computer Science

Computer Science Department

Course no.: 2-7063510-1

Degree: B.A., M.A.

# Schedule

3rd year, 2nd semester.

# Lecturer

Prof. Erel Segal-Halevi, 58.3.26, 09-7431290, davidesh@ariel.ac.il.

Office hours: Thursday 15:00-17:00, by appointment.

# Teaching assistant/tutor

# None

# Course goal

# The course goal is to connect the world of theoretical computer science research with the world of programming. Each year, many new algorithms are developed and published in leading journals; the course aims to enable students to read, program and apply these new algorithms in their programming work. These skills are particularly useful for students aiming for a job in a research & development department in the industry, or for a higher degree in academia.

# Prerequisites

\* Passing grades in: Algorithms 2, Systems Programming 2, Advanced English 2.

\* Basic knowledge of Python is required. The course teaches advanced Python.

\* As the course requires proficiency in both theory and programming, it is recommended only for students who have excellent grades both in theoretical courses and in programming courses.

# Method of Instruction

# The course applies project-based learning. At the beginning of the course, every team chooses a research paper published in a top-level journal or conference in recent years. During the course, every team has to program the algorithms described in their chosen paper in Python; test them thoroughly on both hand-crafted and random inputs; check their performance on inputs of various sizes; suggest and examine various performance improvements; and construct a website for demonstrating their algorithms.

# The material is taught in a weekly lecture, given frontally in class, and simultaneously broadcast in Zoom. The lecture consists of two parts: In the first part, selected students present their solutions to last week’s assignment; in the second, new material is taught.

# Course plan

The plan is tentative and subject to change.

|  |  |  |
| --- | --- | --- |
| Lesson unit | Subject | Lesson # |
| 1 | Introduction: How to read a research paper?  Python: writing short, clear and documented code. | 1 |
| 2 | Python: scientific programming libraries: NumPy, cvxpy, matplotlib, networkx. | 2 |
| 3 | Creating execution examples for new algorithms | 3 |
| 4 | Python development process: logging, virtualenv, pytest, GitHub Actions. | 4 |
| 5 | Python design patterns: decorators, context managers, Strategy, FlyWeight, Command. | 5 |
| 6 | Integrating your algorithms into an open-source library: writing headings, doctests and unittests. | 6 |
| 7 | Python databases: pandas, SQLite, request, google sheet | 7 |
| 8 | Python processes, threads, cython, cppyy, numba. | 8 |
| 9 | Implementation details of algorithms. | 9 |
| 10 | Python: constructing websites for demonstrating your algorithm using Flask. | 10 |
| 11 | Performance analysis: comparing your algorithm to existing algorithms. | 11 |
| 12 | Improving the performance of your algorithm. | 12 |
| 13 | Final project presentations. | 13 |

# Course requirements:

# There will be assignments of two types:

# Python assignments – for rehearsing practicing the material taught in class.

# Rolling assignments – for developing the research algorithm project.

# Course grading:

# The Python assignments are worth 1 point each, for a total of 5%.

# The rolling assignments are worth 10 points each, for a total of 50%.

# Additionally, each student should give 3 presentations during class; each presentation is worth 15%. Total: 45%.

# Learning Outcomes:

By the end of the course, you will be able to:

1. Read and understand a research paper describing new algorithms.

2. Explain and exemplify the algorithms using hand-crafted examples.

3. Program the algorithms in Python.

4. Thoroughly test the algorithms on both hand-crafted and random inputs.

5. Analyze the performance of algorithms in comparison to existing algorithms.

6. Apply various strategies for improving algorithm performance.

7. Construct a website in Flask for demonstrating algorithm operation.

# Main textbook and additional text books:

1. Lecture notes from previous years (in Hebrew) are available at GitHub:

<https://github.com/erelsgl-at-ariel/research-5785>

2. David Kopec: “***Classic Computer Science Problems in Python*”, Manning Publications Co., 2019.**

3. Basnat Agarwal: “***Hands-On Data Structures and Algorithms with Python*”, Packt Publications, 2018.**

**In addition to the textbooks, there is a list of about 100 research papers, from which each team should choose a paper to work on for their project. The list is updated every year and can be found on the course website at the above link.**

# Required material for the examination

# There is no exam – the course grade is based on assignments only.