Engineering Tripos Part IIB, 4M26: Algorithms and Data Structures, 2021-22

Module Leader

Dr I Budvytis [1]

Lecturer

Dr I Budvytis

Timing and Structure

Lent term. 16 lectures (including 3 examples classes). Intake: Part IIB students, Part IIA (optional).

Prerequisites

Due to a novel form of assessment (coding based exam) the intake of this course will be limited to 30-60 students for the first year (2021-2022). Only students who have strong skills in coding with Python are expected to attend this module. Students who are interested in taking this module will be required to pass a coding test in Michaelmas 2021. It is important to have an alternative course in mind for Lent term in case one does not pass the test.

Aims

The aims of the course are to:

- Introduce the principles behind algorithm and data structure design and evaluation.
- Cover key topics including elementary and advanced data structures, sorting algorithms, graph algorithms, etc.
- Provide an extensive hands-on understanding of the aforementioned topics via coding-focused computerised examples papers and exam.

Objectives

As specific objectives, by the end of the course students should be able to:

- Analyse computational efficiency of most algorithms.
- Re-implement and debug algorithms taught under time constraints.
- Correctly choose the right algorithmic solution and data structures for the problem encountered.
- Understand relative theoretical and practical advantages and disadvantages of various methods.
- Devise and implement new algorithms or modify existing algorithms to solve previously unencountered tasks.

Content

- **Introduction (1L)**
  - Algorithms and Data Structures: what is it, why study it and how? Introduction of the coding platform and other resources. Applications.
- **Fundamentals of Algorithms (2L)**
  - Elementary data structures - stacks and ques, linked-lists, arrays, dictionaries. Algorithmic

complexity and NP completeness. Strategies for algorithmic design: divide and conquer, dynamic algorithms, greedy algorithms.

- **Advanced Data Structures (2L)**
  - Hash tables, binary search trees, red-black trees, etc.
- **Sorting Algorithms (2L)**
  - Sorting algorithms - Bubblesort, Heapsort, Quicksort, etc. Sorting in linear time.
- **Graph Algorithms (3L)**
  - Graph algorithms - shortest path (BFS, DFS, Dijkstra, Bellman-Ford, etc), maximum flow (Ford-Fulkerson), minimum spanning trees (Kruskal's, Primm's).
- **Computational Geometry and Mathematical Algorithms (2L)**
  - Line segment intersection, closest pair, convex hull. DFT and FFT, Matrix multiplication and inversion, number theoretic algorithms.
- **Advanced Topics (1L)**
  - Multi-threaded algorithms, approximation algorithms, differentiable programming.
- **Example classes (3L)**
  - Discussion of examples papers and past examination papers.

Examples papers

A computerised exam is held at the Design Project Office (DPO). A mixture of (i) coding, (ii) simple pen-and-paper algorithm run-through and (iii) short theoretical questions are provided in the exam paper. See example question here: http://mi.eng.cam.ac.uk/~ib255/files/4M26-Algorithms_and_Data_Structures... [2]

Booklists

Introduction to Algorithms (3rd ed) *by Cormen, T., Leiserson, C., Rivest, R., Stein, C. The MIT Press. ISBN:978-0-262-03384-8.*

Also, please refer to the Booklist for Part IIB Courses for references to this module, this can be found on the associated Moodle course.

Examination Guidelines

Please refer to Form & conduct of the examinations [3].

Last modified: 04/06/2021 15:57

Links
[1] mailto:ib255@cam.ac.uk
[2] http://mi.eng.cam.ac.uk/~ib255/files/4M26-Algorithms_and_Data_Structures-S1-Jupyter-Notebook.pdf
[3] https://teaching26-27.eng.cam.ac.uk/content/form-conduct-examinations