Survival Guide 02282

Inge Li Gørtz Philip Bille

Welcome to 02282. Here is a short survival guide for the course.

Structure

The course runs on Mondays 8-12. There is a weekplan for each week containing information about literature for the week and exercises. The class will be a mixture of lectures and exercises. Be prepared to participate actively in the class.

Litterature The litterature is a mix of scientific papers, chapters from algorithm books, and lecture notes.

DTU Learn The course uses DTU Learn as online forum for announcements and discussions. It is also here you hand in your mandatory exercises. Remember to set up notifications on DTU Learn, otherwise you might miss important information. Do not use DTU Learn for discussing solutions to mandatory exercises (see collaboration policy).

Mandatory assignments

The course contains approximately bi-weekly mandatory exercises that counts toward a large fraction of the final grade. The final grade is an overall evaluation of your mandatory exercise and the oral exam combined. Thus, there is no precise division of these part in the final grade. However, expect that (in most cases, and under normal circumstances) the mandatory exercises account for a large fraction of the final grade.

- Use the template.tex file to prepare your hand in exercises.
- Do not repeat the problem statement in your hand in.
- Compile using LaTeX. Upload the resulting pdf file (and only this file) via DTU Learn.
- The maximum size of the finished pdf must be at most 2 pages.
- An exercise from week x must be handed in no later than Sunday in week x before 20.00. This is a firm deadline – exercises handed in after the deadline will not be graded.

Study tips

How do I prepare for class? You are expected to read *before* the lecture. On each week plan you can see what you are expected to read for that week. The lecture is meant to clarify and give an overview — it will not necessarily cover all material for the week in details.

How should I write my mandatory exercises? The ideal writing format for mandatory exercises is classical scientific writing, such as the writing found in the peer-reviewed articles listed as reading material for this course (not textbooks and other pedagogical material). One of the objectives of this course is to practice and learn this kind of writing. A few tips:

- Write things directly: Cut to the chase and avoid anything that is not essential. Test your own writing by answering the following question: Is this the shortest, clearest, and most direct exposition of my ideas/analysis/etc.?
- Add structure: Dont mix up description and analysis unless you know exactly what you are doing. For a data structure explain following things separately: The contents of the data structure, how to build it, how to query/update it, correctness, analysis of space, analysis of query/update time, and analysis of preprocessing time. For an algorithm explain separately what it does, correctness, analysis of time complexity, and analysis of space complexity.
- Be concise: Convoluted explanations, excessively long sentences, fancy wording, etc. have no in place scientific writing. Do not repeat the problem statement.
- Try to avoid pseudocode: Generally, aim for human readable description of algorithms that can easily and unambiguously be translated into code.
- Examples for support: Use figures and examples to illustrate key points of your algorithms and data structures.

Can I write my assignments in Danish? Ja. Du er meget velkommen til at aflevere på dansk.

Prerequisites

Undergraduate level courses in algorithms and data structures (comparable to 02105 + 02110) and mathematical maturity.

You should have a working knowledge of algorithm analysis (e.g. asymptotic notation, worst case analysis, amortized analysis, basic analysis of randomized algorithms), data structures (e.g. stacks, queues, linked lists, trees, heaps, priority queues, hash tables, balanced binary search trees, tries), graph algorithms (e.g. BFS, DFS, single source shortest paths, minimum spanning trees, topological sorting), dynamic programming, divide-and-conquer, and NP-completeness (e.g. basic reductions).

If you do not have the prerequisites then you can read up on it in e.g. ”Introduction to Algorithms” by Cormen, Leiserson, Rivest, and Stein, or in another introductory algorithms book.

Collaboration policy for mandatory exercises

- You may collaborate with up to 2 fellow students on the hand in exercises. The collaborators must be listed in your solution (see template).
- Collaboration is limited to discussion of ideas only, and you should write up the solutions entirely on your own.
- Do not use or seek out solutions from previous years of the course, solutions from similar courses, or solutions found on the internet.
- Do not discuss your solution or ideas for solutions publicly on DTU Learn.

Violations of the collaboration policy will be reported.