Institut for Matematik og Datalogi
Syddansk Universitet, Odense

October 8, 2018
KSL

MIDTERM EVALUATION OF DM819 – FALL SEMESTER 2018

Evaluation Form

Notes were made available and students were asked to write at least one statement on a note and put it in a jar.

In a second round, all notes from the jar were passed around and each note was annotated by all other students, using the following symbols, with the stated meaning:

?	÷	(÷)	—	(✓)	✓
What?!	Disagree	Disagree some	Indifferent	Agree some	Agree

Everything was anonymous and the lecturer received all the annotated notes at the end.

Below is a list of all the comments with a count of annotations and the lecturer's remarks, if any.

Due to the similarity of some remarks, I'll categorize them some, but I will list all remarks explicitly so that all students can see that their remarks were not lost. I have fixed typos in a few remarks, but otherwise they are stated verbatim as you wrote them. The order is lectures, the lecturer, exercises, project & code, and lectures at 8.

Thank you for your feedback!

Results

I like the course.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	1	10

See response further down.

The content of the course is interesting.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	0	11

See response further down.

The content of the course is interesting and has a nice balance between practical problems and theory.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	1	10

See response further down.

It's nice to see some "practical" applications of what we learn.

?	÷	(÷)	—	(✓)	✓
0	0	0	1	0	10

I'm glad you like the course so far. I also like the blend of algorithms that are nice from a theoretical viewpoint wrt. correctness and complexity, and at the same time are useful. There's a strong focus on efficient algorithms and I believe everything taught in this course is used in some applications.

The course continues in the same style, so hopefully you'll like the second part as well.

The course is good – actually very nice how you explain things.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	0	11

See response further down.

Lectures are well structured.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	2	9

See response further down.

Kim explains the course content nicely, making it easy to understand.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	0	11

See response further down.

Algorithms (steps & complexity) are well explained during lectures.

?	÷	(÷)	—	(✓)	✓
0	0	0	0	0	11

I'm glad you like how I teach it. Though I hate preparing for lectures, I always do what it takes, since I hate being unprepared even more. ☺ And I

enjoy the actual lecturing and communicating with you students. This is of course to a large extent to your credit, since you're clearly paying attention and being responsive.

I like that the speed is not too fast such that I can easily follow.

?	÷	(÷)	—	(✓)	✓
0	0	0	1	6	4

See response further down.

This course covers a lot of interesting topics, so every time after the lectures, I feel like implementing them. However, some are just too complicated for one week, so it might be useful to slow down the pace of the lectures.

?	÷	(÷)	—	(✓)	✓
0	2	2	3	4	0

I take the annotations to the previous remark to mean that if I can slow down slightly that would maybe be better. On the other hand, this one is pretty evenly divided around the "no change" option. Anyway, I'll definitely keep it in mind and maybe slow down at the more difficult spots.

Lectures are interesting, but exercises are tough.

?	÷	(÷)	—	(✓)	✓
0	0	2	1	6	2

See response further down.

It would take a while to prepare all exercises each week, but the amount of exercise work is still adequate as I feel like I understand them at the exercise session.

?	÷	(÷)	—	(✓)	✓
0	0	0	4	2	5

See response further down.

Too may exercises.

?	÷	(÷)	—	(✓)	✓
0	2	5	2	2	0

Opinions are somewhat divided, but the majority seems to think that there is an adequate number of exercises, but also that they are hard, and that it takes a long time to prepare for them.

Actually, the exercises are quite mixed. Some are easy, and some, admittedly, quite hard. You get most out of the course and the exercise session if you try to solve all of them. With the easy ones, you'll succeed, and with the hard ones, you'll get a better understanding of why they are hard, and you'll learn more from seeing the solution than you would if you didn't attempt solving them yourselves. I don't expect that you'll all be able to solve the hardest exercises, but you still learn from them, and they highlight important aspects. The focus at the exam is on the algorithms and analysis from the main text. You will not be asked to solve exercises, but I believe discussing them helps you understand the properties and limitations of the material covered in the text.

I don't know how much time you spend, but the government's and SDU's official position is that you should work at least 1650 hours per year, which is 275 hours for a 10 ECTS course. On average, you spent 20 hours on the obligatory assignment, part 1, so let's say that the next one will take twice as long. This leaves 215 hours, spread out over at most 14 weeks, so about 15 hours per week. Subtracting the times we meet, this leaves 11 hours for studying the textbook and solving and postprocessing exercises. I know, of course, that just like everyone else, you balance lots of things such as family life, work, transportation, etc., and it's not uncomplicated for everyone to work 45 hours per week, just because you get long vacations.

A bit hard to have the exercise session 2 weeks after the lecture on that topic. Maybe it's good for remembering, though.

?	÷	(÷)	—	(✓)	✓
1	0	0	2	5	2

I don't know what the optimal learn/review interval is. I want you to have at least one week just to make sure to include the slot where you have time to schedule exercise preparation, given your other tasks. I usually aim for 8–10 days, but with our current schedule it's 12–13 days. I'll think about it if a cancelation gives me a chance to reschedule.

It's nice to have a practical project.

?	÷	(÷)	—	(✓)	✓
0	0	0	1	2	8

It highlights very different aspects and I think it's a nice supplement. It's important to be aware that it is a supplement and does not prepare you for the oral exam.

Would like to see actual code examples instead of pseudocode.

?	÷	(÷)	—	(✓)	✓
0	5	2	0	3	1

Opinions are very polarized for this statement. It may be due to interests, different traditions, or (lack of) fondness of algorithms. However, my aim is that you think in terms of loops, conditionals, and functions, as well as data structures such as arrays, linked lists, balanced binary search trees, etc. These concepts can be realized in any programming language and I think of the concrete syntax of a real programming language as clutter that obscures the important ideas. It's not because I don't enjoy programming; ever so often I get an urge to see something realized in practice and I program it. However, when teaching algorithms, I don't find it very useful to look at the many lines of codes necessary to program what can be specified very briefly using pseudocode.

8 o'clock is the best time for lectures.

?	÷	(÷)	—	(✓)	✓
0	3	2	4	1	1

See response further down.

Too early.

?	÷	(÷)	—	(✓)	✓
0	4	0	6	0	1

See response further down.

8 o'clock in the morning is early.

?	÷	(÷)	—	(✓)	✓
0	3	0	6	1	1

See response further down.

I am fine with the classes being at 8 in the morning.

?	÷	(÷)	—	(✓)	✓
0	0	0	4	2	5

It seems only a few people are significantly bothered by the early time. At any rate, I'm very happy that you all show up! I'm here before 7:30 every morning, so I'm fine with the time. However, I fully respect that people have different cycles. I'm not requesting 8 in the morning for my course. The time slots are decided upon by our timetabler who focuses on avoiding conflicts for the students. That said, I do make it clear that I think that if the choice is between 16–18 and 8–10, I think it should be 8–10. This is just because the Danish society is based on 8–16 working hours, which means that when there are students with kids, they have to pick them up shortly after 16:00.