Discrete Mathematics

Welcome

Prof Hans Georg Schaathun

Høgskolen i Ålesund

Autumn 2014 – Week 1

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

The lecturer

- cand.scient. in *industrial and applied mathematics and informatics*
    - specialised in coding theory and cryptography
    - key applications of discrete mathematics
- dr.scient. 2002 in Coding theory
- post.doc. 2003-2006 in coding and cryptography at University of Bergen
- lecturer/senior lecturer 2006-2010 in multimedia security at University of Surrey
- professor HiÅ from 1 February 2011
    - inter- and multi-disciplinary research
    - software engineering

Mathematics and Computing

- Two angles to discrete mathematics
    1. the computer science application
    2. the abstract mathematics
- This module is mathematics and computing
    - not 50–50 — it is 100% of both
- Application driven module
- using abstract thinking and formalism to an end

HØGSKOLEN I ÅLESUND
Aalesund University College

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

Discrete Mathematics

God made the natural numbers; all else is the work of man.

«Die ganzen Zahlen hat der liebe Gott gemacht, alles andere ist Menschenwerk»

Leopold Kronecker (1823-1891)

- Natural numbers are those that you count
  - 1,2,3,4,5,. . .

(Some authors count zero as a natural number.)

Natural numbers

Discrete

1. Natural numbers
2. Countable objects
3. Nothing between 2 and 3

Continuous

1. Real and complex numbers
2. Measurements and approximations
3. Always points in between
   - $(a,b) \leftarrow (a+b)/2$

This is where computing and other engineering disciplines go their separate ways ...

Mathematics for computing

- Digital refer to *digit* meaning *finger*
  - you count on your fingers
- Computers deal with discrete objects
- Finite number of memory states — definately countable
- Floating point numbers give a good approximation to continuous (real) numbers
  - but they are still discrete
- Discrete mathematics is the corner stone for computing

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

Time table and work load

- 10 credits ECTS
  - Expected work load is 250–300h
  - about 16–20h(!) per week
- Three two-hour sessions per week (6h)
  - schedule another 10–14h per week to read, watch video, and prepare
  - at least 2h before each lecture

Sessions

- Three main learning activities
  - video clips – to be watched in *your own time*
  - exercise sheets – to practice skills and test understanding
  - supervised sessions – collaborative problem solving, Q&A
  - student-led tutorials – mandatory

You need to prepare for sessions.

Corrections

- Learning material is not static
- Changes will be made to
  - Correct errors
  - Correct omissions
  - Make material easier to understand
- Changes announced on web page

Make sure you are familiar with the latest version before the exam.

Let me know, if something is missing or not satisfactory.

Textbook

- Two options. Rosen or Stein *et al*
- The textbooks do not define the syllabus.
- Syllabus defined by
  1. exercises
  2. videos

Student-led tutorials

1. Solve assigned problems
    - prepare to present solutions
2. Attend class
    - tick problems you can present on the class list
3. For each problem,
    - a random student is appointed to present the solution
4. To sit the exam, you need 40% problems ticked
5. If you bluff and cannot present when called,
    - all ticks of the day are cancelled

Web page

- `http://www.hg.schaathun.net/DisMath/`
- All material is found here.
- Some is password protected. See fronter for password.
- Backup: `http://kerckhoffs.schaathun.net/DisMath/`

Contact me: hasc@hials.no

> *Remember copyright. Redistribution of the material is prohibited.*

Technical issues

- Video (mpeg4 or ogg/theora)
  - mplayer, vlc, QuickTime
  - should play directly in most browsers (chromium, Safari)
- Web pages with MathML
  - use firefox or iceweasel (not chrome/chromium)
- clean slides (PDF) should work without trouble
- slide notes (PDF with annotations)
  - Firefox plugin may not show annotations
  - Works in Safari, (Mac) preview, okular, skim, etc.

Let me know, if you have any technical problems.

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

Prerequisites

1. What is a set?
2. What is a function?
3. What is a matrix?

Set theory

$$\begin{aligned} S &= \{1,2,3\} \\ T &= \{1,5,a\} \\ S \cup T &= ??? \\ S \cap T &= ??? \\ S \backslash T &= ??? \end{aligned}$$

Function

$$f : A \to B$$
$$f(x) =$$

- domain
- range
- co-domain
- function value *or* return value

Linear algebra

$$
\begin{aligned}
A &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \\
B &= \begin{bmatrix} 0.5 & 0.2 \\ 0.1 & 0 \end{bmatrix} \\
B \cdot A &= ??? \\
B + A &= ??? \\
A^{\mathrm{T}} &= ???
\end{aligned}
$$

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

Pseudo-code

What is pseudo code?

Input Array A of length n

Output The same array A sorted in place.

```
for out_idx := 1 to n-1
    for in_idx := out_idx+1 to n
        if A[out_idx] > A[in_idx]
            swap A[out_idx] with A[in_idx]
```

Pseudo-code

What is pseudo code?

Input Array A of length n

Output The same array A sorted in place.

```
for out_idx := 1 to n-1
    for in_idx := out_idx+1 to n
        if A[out_idx] > A[in_idx]
            swap A[out_idx] with A[in_idx]
```

Pseudo-code

- Explain algorithms
- Human readers
- No standard syntax
- Mix elements from
  - Plain English
  - Well-known programming languages
  - Mathematical notation
- Choose the most readable language

Example of variation

All of these lines express the same thing.

- `if A[out_idx] >= A[in_idx]`
- if `A[out_idx]` $\geq$ `A[in_idx]`
- if $A_i \geq A_j$

The variability and inconsistency will be evident in this module.

Outline

1. Why I teach this module?
2. What is discrete mathematics?
3. The module
4. Prerequisites
5. Pseudo code
6. Exercises

Exercises

- Repetition
  1. Set theory
  2. Functions
- Testing
  - web page; can you access it?
  - videos: do they play on your system?