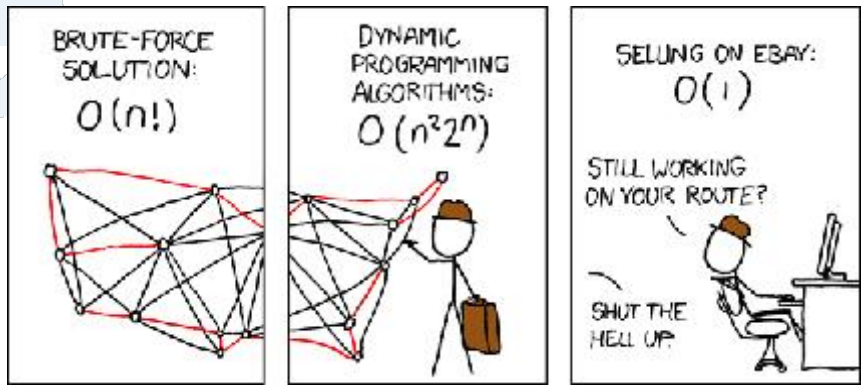
OHJ-2306 Fall 2011
Introduction to Theoretical Computer Science



Organization & timetable

Lectures: prof. Tapio Elomaa, M.Sc. Timo Aho

- Tue and Thu 14–16 TB220
- Aug. 30 – Dec. 8, 2011
- **Exceptions**:
  - Thu Sept. 1 – lecture cancelled due to the illness of the lecturer
  - Tue Sept. 6 & Thu Sept. 8 – lectures given by M.Sc. Timo Aho
  - Thu Oct. 4 – lecture cancelled, ALT/DS 2011 in Espoo
  - Tue Oct. 18 & Thu Oct. 20 – no lectures, period break
  - Tue Dec. 6 – lecture cancelled, independence day

Weekly exercises start later →

- M.sc. Teemu Heinimäki,

Exam: Thu Dec. 15, 2011, 9–12 AM
Mon Jan. 30, 2012, 9–12 AM
Mon Mar. 19, 2012, 9–12 AM

Assessment: 30 + 6 = 36

{tapio.elomaa, timo.aho, teemu.heinimaki}@tut.fi

Weekly exercises

- It is **most advisable** to take part in the weekly exercises
- Being ready to present your solution to problem yields one mark
- Each weekly exercise session has approx. 6 problems → altogether you can collect approx. 6×10 = 60 marks

Marks	Extra points
25% (c. 15)	1
35% (c. 21)	2
45% (c. 27)	3
60% (c. 36)	4
75% (c. 45)	5
85% (c. 51)	6

Grading

- You can pass the course with grade 5/5 by taking the exam alone, but
  - The exam (max 30 p.) is not the easiest of them all
  - Therefore it is advisable to participate in the weekly exercises from which you can earn extra points (max 6 p.)
  - By independently solving the problems you learn the seemingly difficult material of the course
- The course grade will *most probably* be determined according to the following table:

points	15	18	21	24	27
grade	1	2	3	4	5

Course Material

- The course text book in Fall 2011 is

  Michael Sipser: *Introduction to the Theory of Computation*, Second Ed. (International Ed.), Thomson, 2006

- Almost any text book on this topic covers the material that we will go through
- The slides will appear into the Web weekly
  - http://www.cs.tut.fi/kurssit/OHJ-2300/2306.html
  - http://www.cs.tut.fi/~elomaa/teach/2306.html/

- The exam is based on the lectures

Timetable for Lectures

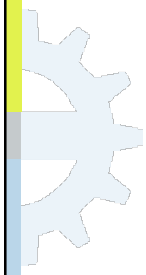
1. **Introduction** (1)
2. **Recap: automata, grammars and languages** (2–4)
   1. Regular languages
   2. Context-free grammars
3. **Computability theory** (4–8)
   1. Universal models of computation
   2. Solvability / decidability
   3. Reducibility
   4. Advanced topics
4. **Complexity theory** (9–14)
   1. Time complexity
   2. Space complexity
   3. Practical solvability
   4. Advanced topics

0. Introduction

- This course offers an introduction to the mathematical and theoretical background of computer science
- Such information is an integral part of the general knowledge of a computer scientist
- The aim is to gain a basic understanding of what kinds of problems can in principle be solved using a computer
- Even more important is to observe which of the decidable problems can be solved *efficiently* by a computer program
- The results that we cover in this course are fundamental in the sense that the increasing efficiency of computers in the years to come will not deem these results insignificant

Future Computers? (from Wikipedia)

- **DNA computing** does not provide any new capabilities from the standpoint of computability theory.
  - For example, if the space required for the solution of a problem grows exponentially with the size of the problem (EXPSPACE problems) on von Neumann machines, it still grows exponentially with the size of the problem on DNA machines.
- **Quantum computing**, on the other hand, does provide some interesting new capabilities.
  - If large-scale quantum computers can be built, they will be able to solve certain problems much faster than any current classical computers (for example Shor's algorithm). Quantum computers don't allow the computations of functions that are not theoretically computable by classical computers, i.e. they do not alter the Church–Turing thesis. The gain is only in efficiency.



and the list goes on …

- Optical computing
- Chemical computing
- Parallel computing
- Cluster computing
- Massive parallel processing
- Distributed computing
- Grid computing
- Cloud computing