Algorithms and Data Structures II

Course 1DL231 (5 credits)

Justin Pearson

Computing Science Division
Department of Information Technology
Uppsala University
Sweden

Period 2, autumn semester:

http://user.it.uu.se/~justin/Hugo/courses/ad2/

Will I Get Rich After Passing This Course?

Some of the wealthiest people on Earth made their fortunes from clever algorithms and data structures, witness:

- Sergey Brin and Larry Page, inventors of the Google search algorithm, valued at many gazillions of SEK.

Some of the wealthiest people in Sweden studied at UU and went on to develop software needing clever algorithms and data structures, witness some of the founders of:

- MySQL, acquired by Sun, and now Oracle for about 6.3 billion SEK.
- Skype, acquired by eBay for at least 17 billion SEK, and now Microsoft
- Trade Extensions, acquired by Coupa for 45 million USD.

There is no limit to the need for clever algorithms and data structures!

Algorithms make you a better programmer

- Often there is a better way of solving a problem than by brute force or simply trying all the possibilities.
- Techniques such as dynamic programming, maximum flow and other graph algorithms allow you to solve common problems.
- During the assignments you will be expected to work out which algorithm should be used for some real world programming task.

Short version, how do I pass difficult coding interview questions. Passing difficult coding interviews is maybe not the most useful thing that you will learn, but the algorithms studied in the course will useful tools to your programming toolbox.

Course Contents

- Reminder: Function growth (O, Θ, Ω, ...); recurrences
- Reminder: Divide & conquer
- Dynamic programming
- Greedy algorithms
- Minimum spanning trees (Prim's algorithm)
- Single-source shortest paths (Dijkstra's algorithm)
- Maximum flow (Ford-Fulkerson algorithm)
- Disjoint-sets data structure
- String matching (Rabin-Karp, Knuth-Morris-Pratt algos)
- Complexity classes P and NP; open question: P $\stackrel{?}{=}$ NP

Learning Outcomes

In order to pass, the student must be able to:

- use the notation of asymptotic growth of functions to describe the complexity of computational problems and algorithms;
- derive equations for the complexity of algorithms, and solve such equations;
- work with common algorithmic techniques such as dynamic programming, greedy algorithms, etc;
- deal with basic problems using graph algorithms, string matching, and flow networks;
- define the complexity classes P and NP, and discuss the open question $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$.

Course Organisation

- Period 2: November to December.
- 13 **lectures**, in English.
- Textbook: *Introduction to Algorithms* (third edition).
  T.H. Cormen, Ch.E. Leiserson, R.L. Rivest, & C. Stein.
  The MIT Press, 2009. (Available online at the library, but one of the best algorithms textbooks that you can buy).
- **Python programming** for 3 **assignments**, (2 credits)
  to be done in pairs if the class is large,
  with $3 \cdot 3 = 9$ **help sessions** and 3 **solution sessions**.
- 1 closed-book **exam**, to be done alone. (3 credits)
- **Prerequisites:** 60 ECTS credits, of which
  min 15 ECTS credits in discrete Mathematics, and
  min 30 ECTS credits in Computer Science, including
  (the equivalent of) *Algorithms and Data Structures I*.

Important Information

This is not just a programming course:

- Report writing is one of the course goals. So producing a good well written report for each assignment is a requirement for passing.
- There is also an emphasis on the reasons why the algorithms are correct. You will be required to do some mathematical reasoning in your report.
- Just submitting code that passes some of the test cases is not sufficient for passing the assignments.

Be warned the assignments will really take 2 credits of your time, on average 30 hours each.

Take a look at the course web-page

Look at http://user.it.uu.se/~justin/Hugo/courses/ad2/ for an idea of the course content and contact me if you have any questions.