Algorithmic Complexity
PGP - CDS Program
Handout

Instructor: Prof. Ashwin Ganesan (`ashwin.ganesan@insofe.edu.in`)

Best coding practices: what this module is about

Suppose you have to write a program to solve a given problem. The following are some rules to follow or criteria to keep in mind:

- choice of programming language(s)
- add comments and indent code; keep code simple and readable
- be consistent in formatting and naming conventions
- does the program meet its specification (ie, does it produce the correct output for each possible input)
- is it extensible? portable?
- are procedures (subroutines) created in such a way that they perform logical subfunctions?
- performance of algorithms (design and analysis of algorithms) - **what this course is about**
  - program efficiency: does the program not only produce the correct output, but also produce the correct output using the least amount of resources? The main resources (costs) are running time and memory space. We shall learn techniques for designing algorithms and data structures which are efficient.
  - what is the performance of the algorithm? We shall learn techniques for analyzing (the performance of) algorithms.

Prerequisites

The prerequisites for this module are basic programming and elementary calculus. More specifically,

- This course assumes familiarity with basic programming. Students are expected to know concepts such as looping and iteration (`while` and `for` loops), branching (if-then-else constructs), and functions, and students should have written simple programs that use these constructs.
  - The first 6 lectures of INSOFE's "CSE 5222: Python for Machine Learning Engineering" can be used as a refresher course in case you haven't done programming in a while.
- Algorithm descriptions in lectures will be in pseudocode. (Read pp. 20-22 of CLRS for pseudocode conventions.)

```
function isEven(x):
    # input: an integer x
    # output: True iff x is even
    if x mod 2 = 0 then
        return True
    else
        return False
```

  - Sometimes a particular programming language such as Python will be used in lectures for illustrative purposes, but completing this course does not require learning the syntax of Python.
  - During the exams, when you are asked to write computer programs, you may write your program either in pseudocode or in a particular programming language.

- There are a few programming assignments in this module. You can write these programs in your favorite programming language. Please feel free to see me if you have any questions on these assignments.

  - It might help to use an IDE (Integrated Development Environment) such as Anaconda or Code::Blocks to do the programming assignments.
  - Anaconda (which comes with Python and the Spyder IDE) can be downloaded for free from https://www.anaconda.com/download/
  - Code::Blocks (also available for free and for both Windows and Linux) can be configured for many languages (C, C++, Java, Python, among others). Website: codeblocks.org

Textbooks

- A. V. Aho, J. E. Hopcroft, and J. E. Ullman, "Data Structures and Algorithms", Pearson India, 2002. [AHU]
- T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, "Introduction to Algorithms", Third edition, Publisher: MIT Press, 2009. [CLRS]

References

- T. Roughgarden, "Algorithms Illuminated, Part 1: The Basics", 2017.
- S. Dasgupta, C. Papadimitriou and U. Vazirani, "Algorithms", Tata McGraw-Hill, 2008.
- D. Knuth, "The Art of Computer Programming, Volume 1: Fundamental Algorithms", Third Edition, Addison-Wesley, 1997.
- D. Knuth, "The Art of Computer Programming, Volume 3: Sorting and Searching", Second Edition, Addison-Wesley, 1998.
- A. Bell, E. Grimson, and J. Guttag. 6.0001, "Introduction to Computer Science and Programming in Python", Fall 2016. MIT OpenCourseWare, https://ocw.mit.edu
- J. Guttag, "Introduction to Computation and Programming Using Python with Application to Understanding Data", Second Edition, MIT Press, 2016.
- J. Leskovec, A. Rajaraman, and J. Ullman, "Mining of Massive Datasets", Third Edition, 2019. Free download from mmds.org.

Topics for each session

#	Topics	Chapter
1	Course information, debugging your code	n/a
2	Time complexity and insertion sort	CLRS, ch. 2
3	Analysis of algorithms, asymptotic notation	CLRS, ch. 3
4	Asymptotic notation, proof of correctness	CLRS, ch. 3
5	Merge sort, analysis of merge sort	CLRS, ch. 2
6	Master theorem	CLRS, ch. 4

Course expectations

This module will require a significant in-class and out-of-class commitment from the student. A student is expected to spend about two hours outside of class for each contact hour (in class).

Learning Management System (LMS)

There is an online portal `https://slms.insofe.com/login/canvas` where slides, assessments, and announcements about this course will be posted. Check this course page regularly (or make sure the automatic email notifications are enabled, so that you receive emails whenever updates are made to this course page.) Feel free to use the discussion forum in this portal to ask questions – both about logistical issues and about the technical subject matter – and to answer each other's questions.