

IMSE 351 Data Structures and Algorithm Analysis

Description Survey of object-oriented design, programming and algorithm analysis techniques; survey of data structures such as lists, stacks, queues, heaps, trees, and graphs. A “neutral” programming language will be used, with occasional reference to C++.

Objectives:

1. Learn how to analyze the efficiency and running speed of algorithms.
2. Understand concepts pertaining to and usage of various data structures.
3. Study a wide variety of algorithms analytically.

Required text:

Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 2003. *Introduction to Algorithms*, 2nd edition. Cambridge, Massachusetts: The MIT Press.

Recommended references:

[Dale, Nell](#). 2003. *C++ Plus Data Structures*, 3rd ed. Sudbury, Massachusetts: Jones and Bartlett Publishers.

Deitel, H. M., and P. J. Deitel. 2005. *C++ How to Program*, 5th ed. Englewood Cliffs, New Jersey: Prentice-Hall, Incorporated.

[Friedman, Frank L.](#), and [Elliot B. Koffman](#). 2003. *Problem Solving, Abstraction, and Design Using C++*, 4th edition. Reading, Massachusetts: Addison Wesley Longman, Incorporated.

Course Outline:

1. Chapter 1 – Overview – the importance of efficient algorithms.
2. Chapter 2 – Getting started with basic analysis of insertion sort and selection sort.
3. Chapter 3 – Growth of functions.
4. Chapters 6-8 – Sorting: heapsort, Shell sort, bubble sort, quicksort, sorting in linear time.
5. Chapters 10-12 – Basic data structures: stacks, queues, linked lists, hash tables, binary search trees (omit 11.5 and 12.4).
6. Chapters 22-23 – Definitions pertinent to graphs, basic graph algorithms and their practical importance.
7. If time permits, a peek into chapter 34.

Grading Administration:

Assignments	30%
Midterm examination (9 March)	30%
Final examination (27 April)	35%
Participation	5%

Statement from Office of the Provost on [academic ethics](#). Specifically, this class runs under the [honor code](#) governing all College of Engineering classes.