ECE 3652: Fundamentals of Computer Engineering

An introduction to fundamental techniques in computer engineering used throughout the graduate curriculum. Covers basic programming and analysis methods. Covers the formulation and solution of a wide range of computer engineering problems, with emphasis on applications from computer-aided-design, parallel computing, fault tolerant computing, computer networks, distributed processing, computer architecture, signal processing, robotics, fault-tolerant computing and coding. The applications of algorithm analysis and complexity theory to analyzing and solving problems will also be discussed.

Instructor: Waleed Meleis, 320 Dana Building, 373-3023, meleis@ece.neu.edu.

Text: Algorithms in C, Third Edition, by Robert Sedgewick, Parts 1-4, and Part 5.

All course material, including handouts, homework, reading assignments, and notices will be posted on the course website: http://www.ece.neu.edu/courses/ece3652

You should check the website regularly. In addition, I will maintain a course mailing list. Students taking the course on the network should send me an email message within the first week of class. Network students should also read the "Information for students taking the course on the network" document that is on the website.

Policies

Grading: Homework: 40%, Midterm exam: 30%, Final exam: 30%.

Students must adhere to the University Academic Honesty and Integrity Policy. Any student who violates this policy will be reported to the university's director of judicial affairs.

In particular, the following are not allowed:

- Intentionally or knowingly representing the words or ideas of another as one's own in any academic exercise without providing proper documentation of source by way of a footnote, endnote, or intertextual note.
- Intentionally using or attempting to use unauthorized materials, information or study aids in any academic exercise.

Detailed course description

Analysis of algorithms: Correctness, empirical analysis, big-oh notation, growth of functions, best-case, average and worst-case analysis. Review of sorting (sequential, binary sort, selection sort, insertion sort).

Data structures: Arrays, linked lists, heaps, strings, and compound data structures. Abstract data types (stacks and queues). Introduction to sets.

Trees and recursion: Review of recursion. Divide and conquer (Tower of Hanoi, merge sort, quicksort), dynamic programming (knapsack). Tree terminology, properties of binary trees. Tree traversal: preorder, inorder, postorder, depth-first, breadth-first.

Introduction to Graphs: Graph terminology, graph representations. Elementary operations on graphs.

Graph search: Depth-first search (finding Eulerian tours, spanning trees, cycles) Breadth-first search (finding shortest paths Search in directed graphs (topological sorting) Set theory (relations, transitive closure, partial order, total order)

Minimum spanning trees: Prim, Kruskal algorithms

Shortest Paths: Source-sink, single-source, all-pairs. Dijkstra, Bellman-Ford algorithms.

Network Flow: Algorithms for maximum flow.

Introduction to computational complexity: Equivalent problems, reductions, infeasible problems, P, NP, NP-completeness