Aims

This module aims to provide an introduction to Set Algebra, Combinatorics, Probability Theory, Graph Theory, Recurrence Relations, and Matrices, for students on Computer Science and Mathematics degree courses.

Syllabus


Probability Theory: Probability Space, Conditional Probability, Independence and Bayes Theorem, Random Variables and Distributions, Expected Value, Variance, Examples of applications to algorithms (5 lectures)

Graph Theory 1: Introduction to graphs, Basic graph terminology, Adjacency Matrices, Paths and connectivity, Connected components, Shortest path problems in weighted graphs, Dijkstra’s Algorithm. (5 lectures)

Graph Theory 2: Trees and spanning trees, Breadth-first search, Kruskal’s and Prim’s Algorithms for a minimal spanning tree, Euler and Hamilton paths, Fleury’s Algorithm for constructing Euler circuits, Estimates for Hamilton circuits. (6 lectures)

Recurrence Relations: Solving problems by iteration, First and second order recurrence relations, Recurrences in Algorithms (3 lectures)

Matrices and Linear Transformations: Linear equations and elementary row operations, Elementary matrices and Gaussian elimination, Echelon matrices, Eigenvectors and eigenvalues, Diagonalization, The rank theorem. (8 lectures)

Teaching and Assessment

Contact Hours: 3 lectures and 1 tutorial per week
Assessment: 30% by class tests or other continuous assessment 70% by end of course 2-hour exam
Resit Type: exam

Content: September 2013
By the end of the course, students should be able to:

- Understand the basic terminology of set theory, graph theory, linear algebra and probability theory
- Understand how formal mathematical objects like sets, graphs, matrices, recurrence relations arise in computer science related problems
- Solve elementary counting problems,
- Solve systems of linear equations
- Apply graph algorithms
- Solve simple recurrence relations
- Compute probabilities
- Appreciate the power of mathematical formalisation, facilitated by the use of precise definitions and notations, in solving practical problems.
- Appreciate the value of careful, quantitative reasoning in analysing problems related to computer science and to recognise that the outcome of such reasoning can defy naive intuition