Ramakrishna Mission Vivekananda Educational and Research Institute

(Formerly Ramakrishna Mission Vivekananda University)

Deemed-to-be-univeristy by Govt India u/s 3 of UGC Act 1957

School of Mathematical Sciences

Department of Computer Science

MSc Computer Science
Scheme of Instructions

(*as on* 02 May 2018)

PO Belur Math, District Howrah

West Bengal 711202

India

Table of Contents(*)

(*) The courses offered for a particular MSc curriculum contains a subset of these courses. Each of these courses have been taught in the MSc curriculum at some point in time.

Discrete Structures, Logic and Computability (CS100)

Credits: 5

Total hours of instruction: 75

Evaluation: 100% Theory

Discrete Structures:
Proofs introduction

Discrete sets, ordered structures, graphs and trees,

Discrete functions, constructing functions, properties of functions, countability, inductive definitions of sets, recursive functions and procedures, grammars

Binary relations, properties of relations, equivalences, partial orders, inductive proofs

Analysis of algorithms, summation, permutation and combinations, probability, solving recurrences, lower bounds, upper bounds, tight bounds on functions (big-oh, big-omega, big-theta, little-oh, little-omega)

Elementary Logic:

Propositional calculus, well formed formulae and semantics, logical equivalences, truth functions and normal forms, addequate set of operators

Formal Reasoning, proof rules, proofs, derived rules, theorems, soundness and completeness,

Formal axiom systems

Predicate Logic:

First order predicate calculus, predicates, quantifiers, well formed formulae with quantifiers, interpretations and semantics, validity

Equivalent formulae, logical equivalences, normal forms, formazlizing natural language sentences

Formal proofs in predicate calculus, universal instantiation, existential generalization, existential instantiation, universal generalization

Applied Logic:

Equality, program correctness and higher order logic

Computability:

Halting problem, undecidability, intractibility

References:

1. James L. Hein: Discrete Structures, Logic and Computability, Jones and Bratlett Pub. 2010
2. J. L. Mott, A. Kandel and T. P. Baker: Discrete Mathematics for Computer Scientists, Reston, Virginia, 1983.
3. 2. D. F. Stanat and D. E. McAllister: Discrete Mathematics in Computer Science, Prentice Hall, Englewood Cliffs, 1977.
4. C. L. Liu: Elements of Discrete Mathematics, 2nd ed., McGraw Hill, New Delhi, 1985.
5. R. A. Brualdi: Introductory Combinatorics, North-Holland, New York, 1977.
6. 5. Reingold et al.: Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, 1977.
7. J. A. Bondy and U. S. R. Murty: Graph Theory with Applications, Macmillan Press, London, 1976.
8. N. Deo: Graph Theory with Applications to Engineering and Computer Science, Prentice Hall, Englewood Cliffs, 1974.
9. E. Mendelsohn: Introduction to Mathematical Logic, 2nd ed. Van-Nostrand, London, 1979.
10. L. Zhongwan: Mathematical Logic for Computer Science, World Scientific, Singapore, 1989.
11. F. S. Roberts: Applied Combinatorics, Prentice Hall, Englewood Cliffs, NJ, 1984.
12. Lewis and Papadimitriou: Elements of Theory of Computation (relevant chapter on Logic), Prentice Hall, New Jersey, 1981.
13. C. L. Chang and R. C. T Lee: Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York and London, 1973.
14. H. Enderton: A Mathematical Introduction to Logic, Academic Press, London, 1972.
15. M. Fitting: First-order Logic and Automated Theorem Proving, Springer, Berlin, 1990.
16. H. Gallier: Logic for Computer Science, John Wiley and Sons, New York, 1987.
17. G.E. Hughes and M.J. Cresswell: A New Introduction to Modal Logic Symbolic Logic, Routledge, 1996.
18. E. Mendelson: Introduction to Mathematical Logic, Van Northand, London, 1979.
19. A Nerode and R.A. Shore: Logic for Applications, Springer, Berlin, 1993.
20. V. Sperschneider and G. Antonio: Logic: A Foundation for Computer Science, Addison-Wesley, California, 1991.
21. I.S. Torsun: Foundations of Intelligent Knowledge-Based Systems, Academic Press, New York, 1995.
22. L.Zhongwan: Mathematical Logic for Computer Science, World Scientific, Singapore, 1989.

Discrete Structures and Logic (CS204)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Discrete Structures:
Proofs introduction

Discrete sets, ordered structures, graphs and trees,

Discrete functions, constructing functions, properties of functions,
countability, inductive definitions of sets, recursive functions and procedures,
grammars

Binary relations, properties of relations, equivalences, partial orders, inductive proofs

Analysis of algorithms, summation, permutation and combinations, probability, solving recurrences,
lower bounds, upper bounds, tight bounds on functions (big-oh, big-omega, big-theta, little-oh, little-

omega)

Elementary Logic:

Propositional calculus, well formed formulae and semantics, logical equivalences,
truth functions and normal forms, addequate set of operators

Formal Reasoning, proof rules, proofs, derived rules, theorems,
soundness and completeness,

Formal axiom systems

Predicate Logic:

First order predicate calculus, predicates, quantifiers, well formed formulae with quantifiers,
interpretations and semantics, validity

Equivalent formulae, logical equivalences, normal
forms, formazlizing natural language sentences

Formal proofs in predicate calculus, universal instantiation, existential generalization,
existential instantiation, universal generalization

References:

1. James L. Hein: Discrete Structures, Logic and Computability, Jones and Bratlett Pub. 2010
2. James L. Hein: Discrete Structures, Logic and Computability, Jones and Bratlett Pub. 2010
3. J. L. Mott, A. Kandel and T. P. Baker: Discrete Mathematics for Computer Scientists, Reston, Virginia, 1983.
4. 2. D. F. Stanat and D. E. McAllister: Discrete Mathematics in Computer Science, Prentice Hall, Englewood Cliffs, 1977.
5. C. L. Liu: Elements of Discrete Mathematics, 2nd ed., McGraw Hill, New Delhi, 1985.
6. R. A. Brualdi: Introductory Combinatorics, North-Holland, New York, 1977.
7. 5. Reingold et al.: Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, 1977.
8. J. A. Bondy and U. S. R. Murty: Graph Theory with Applications, Macmillan Press, London, 1976.
9. N. Deo: Graph Theory with Applications to Engineering and Computer Science, Prentice Hall, Englewood Cliffs, 1974.
10. E. Mendelsohn: Introduction to Mathematical Logic, 2nd ed. Van-Nostrand, London, 1979.
11. L. Zhongwan: Mathematical Logic for Computer Science, World Scientific, Singapore, 1989.
12. F. S. Roberts: Applied Combinatorics, Prentice Hall, Englewood Cliffs, NJ, 1984.
13. Lewis and Papadimitriou: Elements of Theory of Computation (relevant chapter on Logic), Prentice Hall, New Jersey, 1981.

Concepts of Programming Languages (CS123)

Credits: 5

Total hours of instruction: 75

Evaluation: 80% Theory, 20% Lab assignments

Introduction to Programming Language
The Origins of Programming Languages, Abstractions in Programming Languages, Computational

Paradigms, Language Definition & Translation

Language Design Criteria
Historical Overview, Efficiency, Regularity, Security, Extensibility, Other design principles, C++: An Object-Oriented Extension of C,
Syntax
Lexical Structure of Programming Languages, Context-Free Grammars and BNFs, Parse Trees and Abstract Syntax Trees, Ambiguity Associativity and Precedence, EBNFs and Syntax Diagrams, Parsing Techniques and Tools, Lexics vs Syntax vs Semantics,
Basic Semantics
Attributes Binding and Semantic Functions, Declarations Blocks and Scope, The Symbol Table, Name Resolution and Overloading, Allocation Lifetimes and the Environment, Variables and Constants, Aliases Dangling References and Garbage
Data Types
Data Types and Type Information, Simple Types, Type Constructors, Type Nomenclature in Sample Languages, Type Equivalence, Type Checking, Type Conversion, Polymorphic Type Checking, Explicit Polymorphism
Control I—Expressions and Statements
Expressions, Conditional Statements and Guards, Loops and Variations on WHILE, The GOTO Controversy and Loop Exits, Exception Handling
Control II—Procedures and Environments
Procedure Definition and Activation, Procedure Semantics, Parameter-Passing Mechanisms, Procedure Environments Activations and Allocation, Dynamic Memory Management, Exception Handling and Environments
Functional Programming
Programs as Functions, Functional Programming in Imperative Language, Delayed Evaluation, Haskell—A Fully Curried Lazy, Language with Overloading, The Mathematics of Functional Programming: Lambda, Calculus
Logic Programming
Logic and Logic Programs, Horn Clauses, Resolution and Unification, The Language Prolog, Problems with Logic Programming, Curry: A Functional Logic Language
Object-Oriented Programming
Software Reuse and Independence, Java, C++, Python, Design Issues in Object-Oriented Languages, Implementation Issues in Object-Oriented Languages,

If time permits, we will cover the following as well:
Abstract Data Types and Modules
The Algebraic Specification of Abstract Data Types, Abstract Data Type Mechanisms and Modules, Separate Compilation in C C++ Namespaces and Java, Packages, Ada Packages, Modules in ML, Modules in Earlier Languages, Problems with Abstract Data Type Mechanisms, The Mathematics of Abstract Data Types
Parallel Programming
Introduction to Parallel Processing, Parallel Processing and Programming Languages, Threads, Semaphores, Monitors, Message Passing, Parallelism in Non-Imperative Languages,

References:

1. Programming Languages Principles and Practice (2 nd /3 rd Edition) – Kenneth C. Louden
2. Concepts in Programming Languages – John C. Mitchell

Oracle Lab (CS 128)

Credits: 2

Total hours of instruction: 30

Evaluation: 100% Lab assignments + Term Project

1. Oracle 12 concepts and architecture (10 hours)
    a. Instance, data-blocks, configuration.
    b. Physical and logical storage structures, data-files.
    c. Memory structures.
    d. Schema objects, table-space, data dictionary.
    e. Database and schema object creation, startup and shut down operations
    f. Tools –SQL developer, SQL * PLUS.
2. Relational algebra operations on Oracle (10 hours)
    a. Create, select, update, delete and variations.
    b. SQL operators
    c. Constraints
    d. Rollback and Commit
    e. Oracle functions – scientific, numerical, string, conversion, date, miscellaneous.
    f. Group functions.
    g. Join and Set operations.
3. Advanced queries on Oracle (10 hours)
    a. Sub-query
    b. Correlated sub-query
    c. Nested Query
    d. Views and updatable views
    e. Indexes and sequences.
4. Procedural programming on Oracle (15 hours)
    a. PL/SQL concepts and constructs
    b. Procedures and functions
    c. Control structures
    d. Exception handling
5. Advanced procedural programming on Oracle (10 hours)
    a. Simple cursors
    b. Cursor for update
    c. Stored procedures, functions and packages.
    d. Triggers – statement and row; before and after.

6. Oracle Security and privileges and Oracle Reports (5 hours)

Suggested References

1. Oracle 12c documentation: https://docs.oracle.com/database/121/index.htm

Introduction to Automata Theory (CS 200)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Automata and Languages: Finite automata, regular languages, regular expressions, equivalence of deterministic and nondeterministic finite automata, minimization of finite automata , closure properties, Kleene’s theorem, pumping lemma and its application, Myhill-Nerode theorem and its uses; Context-free grammars, context-free languages, Chomsky normal form, closure properties, pumping lemma for CFL, push down automata.

Computability: Computable functions, primitive and recursive functions, universality, halting problem, recursive and recursively enumerable sets, parameter theorem, diagonalization, reducibility, Rice's Theorem and its applications. Turing machines and variants; Equivalence of different models of computation and Church-Turing thesis.

Complexity: Time complexity of deterministic and nondeterministic Turing machines, P and NP, NP-25completeness, Cook's Theorem, other NP – Complete problems.

References:

1. J. E. Hopcroft, J. D. Ullman and R. Motwani: Introduction to Automata Theory, Languages and Computation, Addison-Wesley, California, 2001.
2. M. D. Davis, R. Sigal and E. J. Weyuker: Complexity, Computability and Languages, Academic Press, New York, 1994.
3. N. J. Cutland: Computability: An Introduction to Recursive Function Theory, Cambridge University Press, London, 1980.
4. H. R. Lewis and C. H. Papadimitriou: Elements of The Theory of Computation, Prentice Hall,Englewood Cliffs, 1981.
5. M. Sipser: Introduction to The Theory of Computation, PWS Pub. Co., New York, 1999.
6. M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to The Theory of NP-Completeness, Freeman, New York, 1979

Finite Mathematics (CS 204)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Recurrence problems, towers of Hanoi, lines in the plane, Josephus problem

Sums, notation, manipulation, multiple sums, generic methods, finite and infinite calculus

Integer functions, floors and ceilings, their applications and recurrences, mod operator, floor/ceil sums

Number theory, divisibility, primes, factorial factors, relative primality

Binomial coefficients, generating functions, hypergeometric functions, hypergeometric transformations

Special numbers, Stirling numbers, Eulerian numbers, harmonic numbers, Bernoulli numbers, Fibonacci numbers

Generating functions and solving recurrences, convolutions

Discrete probability, means and variances, probability generating functions

Asymptotics, big-oh notation

References:

1. Ronald L. Graham; Donald E. Knuth; Oren Patashnik: Concrete Mathematics, Pearson Education, 1994
2. James L. Hein: Discrete Structures, Logic and Computability, Jones and Bratlett Pub. 2010
3. J. L. Mott, A. Kandel and T. P. Baker: Discrete Mathematics for Computer Scientists, Reston, Virginia, 1983.
4. 2. D. F. Stanat and D. E. McAllister: Discrete Mathematics in Computer Science, Prentice Hall, Englewood Cliffs, 1977.
5. C. L. Liu: Elements of Discrete Mathematics, 2nd ed., McGraw Hill, New Delhi, 1985.
6. R. A. Brualdi: Introductory Combinatorics, North-Holland, New York, 1977.
7. 5. Reingold et al.: Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, 1977.
8. J. A. Bondy and U. S. R. Murty: Graph Theory with Applications, Macmillan Press, London, 1976.
9. N. Deo: Graph Theory with Applications to Engineering and Computer Science, Prentice Hall, Englewood Cliffs, 1974.
10. E. Mendelsohn: Introduction to Mathematical Logic, 2nd ed. Van-Nostrand, London, 1979.
11. L. Zhongwan: Mathematical Logic for Computer Science, World Scientific, Singapore, 1989.
12. F. S. Roberts: Applied Combinatorics, Prentice Hall, Englewood Cliffs, NJ, 1984.
13. Lewis and Papadimitriou: Elements of Theory of Computation (relevant chapter on Logic), Prentice Hall, New Jersey, 1981.

Introduction to Probability (CS 205)

Credits: 3

Total hours of instruction: 45

Evaluation: 100% Theory

1. Basic Probability
   a. Introduction
   b. Sample Spaces
   c. Probability Measures
   d. Computing Probabilities: Counting Methods
      i. The Multiplication Principle
      ii. Permutations and Combinations
   e. Conditional Probability
   f. Independence
2. Random Variables
   a. Discrete Random Variables
      i. Bernoulli Random Variables
      ii. The Binomial Distribution
      iii. Geometric and Negative Binomial Distributions
      iv. The Hypergeometric Distribution
      v. The Poisson Distribution
   b. Continuous Random Variables
      i. The Exponential Density
      ii. The Gamma Density
      iii. The Normal Distribution
      iv. The Beta Density

    c. Functions of a Random Variable
3. Joint Distributions
    a. Introduction
    b. Discrete Random Variables
    c. Continuous Random Variables
    d. Independent Random Variables
    e. Conditional Distributions
        i. The Discrete Case
        ii. The Continuous Case
    f. Functions of Jointly Distributed Random Variables
        i. Sums and Quotients
        ii. The General Case
4. Expected Values
    a. The Expected Value of a Random Variable
        i. Expectations of Functions of Random Variables
        ii. Expectation of Linear Combinations of Random Variables
    b. Variance and Standard Deviation
    c. Covariance and Correlation
    d. Conditional Expectation
    e. Definitions and Examples
    f. The Moment-Generating Function
5. Limit Theorems
    a. Introduction
    b. The Law of Large Numbers
    c. Convergence in Distribution and the Central Limit Theorem

References:

1. Introduction to time series analysis; PJ Brockwell and RA Davis
2. Time Series Analysis and Its Applications; Robert H. Shumway and David S. Stoffer
3. Introduction to Statistical time series; WA Fuller
4. A first course in Probability, Sheldon Ross, Pearson Education, 2010
5. Time Series Analysis; Wilfredo Palma
6. P. G. Hoel, S. C. Port and C. J. Stone: Introduction to Probability Theory, University Book Stall/Houghton Mifflin, New Delhi/New York, 1998/1971.

Probability and Stochastic Process (CS 206)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

1. Basic Probability
    a. Introduction
    b. Sample Spaces
    c. Probability Measures
    d. Computing Probabilities: Counting Methods
        i. The Multiplication Principle
        ii. Permutations and Combinations
    e. Conditional Probability
    f. Independence
2. Random Variables

a. Discrete Random Variables
    i. Bernoulli Random Variables
    ii. The Binomial Distribution
    iii. Geometric and Negative Binomial Distributions
    iv. The Hypergeometric Distribution
    v. The Poisson Distribution
b. Continuous Random Variables
    i. The Exponential Density
    ii. The Gamma Density
    iii. The Normal Distribution
    iv. The Beta Density
c. Functions of a Random Variable

3. Joint Distributions
    a. Introduction
    b. Discrete Random Variables
    c. Continuous Random Variables
    d. Independent Random Variables
    e. Conditional Distributions
        i. The Discrete Case
        ii. The Continuous Case
    f. Functions of Jointly Distributed Random Variables
        i. Sums and Quotients
        ii. The General Case
4. Expected Values
    a. The Expected Value of a Random Variable
        i. Expectations of Functions of Random Variables
        ii. Expectation of Linear Combinations of Random Variables
    b. Variance and Standard Deviation
    c. Covariance and Correlation
    d. Conditional Expectation
    e. Definitions and Examples
    f. The Moment-Generating Function
5. Limit Theorems
    a. Introduction
    b. The Law of Large Numbers
    c. Convergence in Distribution and the Central Limit Theorem
6. Stochastic Process
    a. Markov chain
        i. State transition matrix
        ii. Hitting time
        iii. Different States
    b. Poisson process

References:

1. Introduction to time series analysis; PJ Brockwell and RA Davis
2. Time Series Analysis and Its Applications; Robert H. Shumway and David S. Stoffer
3. Introduction to Statistical time series; WA Fuller
4. Time Series Analysis; Wilfredo Palma

Graph Theory and Algorithms(CS 211)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% Programming assignments

1. Network flows:

Max-flow problem and the Ford-Fulkerson algorithm, Design and analysis.
max-flow min-cut theorem statement,proof and analysis
Improvements: Preflow-push, Edmonds, Dinics, MPM, with analysis.

2. Application of Network Flows.

Bipartite Matching, Circulation with demands, Survey Design,Airline Scheduling, Image Segmentation

3. Coloring

Brooks theorem, Erdos' result on chromatic number, Vertex coloring and hardness, list coloring, planar graphs,

4. Geometric structures using graphs:

Interval trees Segment Trees, Range trees, Hilbert Trees.

5. Graph Algorithms

Spanning trees, Biconnected Components, SSP-Shortest path(Dijkstra), APSP-shortest path (Bellman-Ford), Hall's Theorem, Matching in general graphs (Edmonds blossom shrinking) with full analysis. Stable Matching with fairness study, Tuttes' condition for matching and application, Max-weighted bipartite matching, Vertex and edge-connectivity

6. Discharging Technique for studying properties of planar graphs.

References:

1. Algorithm Design. Kleiberg and Tardos, Pearson Publication

2. Introducton to Graph theory, Douglas West, PHI

Computational Geometry (CS 212)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Polygons, Triangulation Theory, Polygon Partitioning, Intersection, Convex Hull, Orthogonal Range Searching, Proximity, Voronoi Diagram, Visibility algorithms, Euclidean Shortest Path, Visibility Graph, Art Gallery Theorem.

References:

1. J. O'Rouke, Computational Geometry in C, Cambridge University Press, 1998.

2. F. Preparata and M. Shamos, Computataional Geometry, Springer, 1985
3. Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, Computational Geometry: Algorithms and Applications, Springer, 2008.

Computer Graphics (CS 214)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% Programming assignments

Introduction to OpenGL

Geometric data types,
vectors and bases, linear transformations, rotations and scaling
World, object and eye frame of references

OpenGL pipeline, OpenGL primitives and
shading language primitives, OpenGL programming fundamentals and sample program using vertex and fragment shaders

Color theory, vision basics, science, hysiology and psychology of vision, implementing color and transparency in OpenGL programs

Projections, orthogonal projection, perspective vision, formulation and implementation

Materials and textures, mipmaps, theory and implementation

Lighting, modelling, point lights, area lights, ray lights, implementing in shaders

Introduction to ray tracing

Shadows, different techniques and algorithms, implementation in OpenGL

Reflections and different algorithms for achieving reflective surfaces

Skyboxes and their implementation in OpenGL

Animation introduction, modelling

Particle effects, fog, fire, smoke, waves, etc.

References:
1. W. K.Giloi: Interactive Computer Graphics: Data Structure, Algorithms, Languages, Prentice Hall, Englewood Cliffs, 1978.
2. W. M. Newman and R. F. Sproull: Principles of Interactive Computer Graphics, McGraw Hill, New Delhi, 1979.
3. J. D. Foley et al.: Computer Graphics, 2nd ed., Addison-Wesley, Reading, Mass., 1993.
4. D. Hearn and P. M. Baker: Computer Graphics, 2nd ed. Prentice Hall of India, New Delhi, 1997.
5. F. S. Hill: Computer Graphics, McMillan, New York,1990.

6. D. P. Mukherjee: Fundamentals of Computer Graphics and Multimedia, Prentice Hall of India, New Delhi, 1999.

Data and File Structures (CS 220)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% programming assignments

Introduction to complexity of algorithms and different asymptotic notations.
Data structure: Introduction to ADT, formal definitions, implementations of basic data structures, array, stack, queue, dequeue, priority queue, linked list, binary tree and traversal algorithms, threaded tree, m-ary tree, heap
Searching: Binary search, Fibonacci search, binary search tree, height balanced tree, splay tree, Red-black tree, digital search tree, trie, hashing techniques.
Records and files: Fixed length/variable length records, pinned/unpinned records, heap file, hashed file, indexed file, relative file, file with dense index, multi-key access file, inverted list, multi-list organization, B-tree, B $*$ -tree, 2-3 tree,

References:
1. T. A. Standish: Data Structures, Algorithms and Software Principles, Addison-Wesley, Reading, Mass., 1995.
2. L. Nyhoff, C++ – An Introduction to Data Structures, Prentice Hall, Englewood Cliffs, 1998.
3. A. M. Tannenbaum and M. J. Augesestein: Data Structures Using PASCAL, Prentice Hall, New Jersey, 1981.
4. D. E. Knuth: The Art of Computer Programming. Vol. 1, 2nd ed. Narosa/Addison-Wesley, New Delhi/London, 1973.
5. T. A. Standish: Data Structure Techniques, Addison-Wesley, Reading, Mass., 1980.
6. E. Horowitz and S. Sahni: Fundamentals of Data Structures, CBS, New Delhi, 1977.
7. R. L. Kruse: Data Structures and Program Design in C, Prentice Hall of India, New Delhi, 1996.
8. A. Aho, J. Hopcroft, and J. Ullman: Data Structures and Algorithms, Addison-Wesley, Reading, Mass., 1983.
9. B. Salzberg: File Structures: An Analytical Approach, Prentice Hall, New Jersey, 1988.
10. T. Harbron: File System Structure and Algorithms, Prentice Hall, New Jersey, 1987.
11. P. E. Livadas: File Structure: Theory and Practice, Prentice Hall, New Jersey, 1990.
12. T. Coreman, C. Leiserson and R. Rivest: Introduction to Algorithms, McGraw Hill, New York, 1994.
13. S. Sahani: Data Structure, Algorithms and Applications in JAVA, McGraw Hill, New York, 2000.
14. D. Wood: Data Structure, Algorithms and Performance, Addison-Wesley, Reading, Mass., 1993.

Introduction to Algorithms (CS 220, CS 221)
Design and Analysis of Algorithms (CS 221)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% programming assignments

1. Different order notations like O, Θ, Ω, θ, ω and compare two different functions using order notation.
2. Methods to calculate and state running time of algorithms using order notations.
3. Divide and Conquer paradigm of algorithm design through its application in devising algorithms for merge sort, counting inversions, finding closest pair of points in a plane, fast integer multiplication, fast Fourier transform etc.
4. Dynamic Programming and use of memoization through several examples like longest increasing subsequence, edit distance, knapsack, matrix chain multiplication, independent sets in trees etc.
5. Greedy methods of algorithm design through various examples like minimum spanning trees, huffman codes, horn clauses etc.
6. Breadth First Search (BFS) and Depth First Search (DFS) in graphs.
7. Application of BFS and DFS like topological sorting of a directed acyclic graph, finding all strongly connected components of a directed Graph, finding articulation points,bridges and biconnected component of a graph, finding Eulerian tour in a Eulerian graph.
8. Kruskal and Prim's algorithm for minimum spanning trees and union find data structure.
9. Algorithms for single source shortest paths in a directed graph like Bellman-Ford algorithm, Dijkstra's algorithm.
10. Algorithms for all pair shortest paths like the matrix multiplication based procedure, Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs.
11. Complexity class NP, NP-Completeness, NP-Hardness, reducibility.

References:

1. Jon Kleinberg, Eva Tardos; Algorithm Design, Pearson education, 2008
2. A. Aho, J. Hopcroft and J. Ullman; The Design and Analysis of Computer Algorithms, A.W.L, International Student Edition, Singapore, 1998
3. S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani; Algorithms, Tata McGraw-Hill c2008
4. S. Baase: Computer Algorithms: Introduction to Design and Analysis, 2nd ed., Addison-Wesley, California, 1988.
5. T. H. Cormen, C.E. Leiserson and R.L.Rivest: Introduction to Algorithms, Prentice Hall of India, New Delhi, 1998.
6. 4. E. Horowitz and S. Sahni: Fundamental of Computer Algorithms, Galgotia Pub./Pitman, New Delhi/London, 1987/1978.
7. K. Mehlhorn: Data Structures and Algorithms, Vol. 1 and Vol. 2, Springer-Verlag, Berlin, 1984.
8. A. Borodin and 1. Munro: The Computational Complexity of Algebraic and Numeric Problems, American Elsevier, New York, 1975.
9. D.E. Knuth: The Art of Computer Programming, Vol. 1, Vol. 2 and Vol. 3. Vol. 1, 2nd ed., Narosa/Addison-Wesley, New Delhi/London, 1973; Vol. 2: 2nd ed., Addison-Wesley, London, 1981; Vol. 3: Addison-Wesley, London, 1973.
10. S. Winograd: The Arithmetic Complexity of Computation, SIAM, New York, 1980.

Compiler Design (CS 226)

Credits: 4

Total hours of instruction: 60

Evaluation: 60% Theory, 40% programming assignments

Introduction: Compiler, phases and passes, bootstrapping, finite state machines and regular expressions and their applications to lexical analysis, implementation to lexical analysers, lexical-analyser generator; LEX-compiler, formal grammars, and their application to syntax analysis, BNF notation, ambiguity, LL(k) and LR(k) grammar, bottom-up and top-down parsers, operator precedence, simple precedence, recursive descent and predictive parsers, LR(k) parsers, parse table generation, YACC.

Syntax directed translation: Quadruples, triples, 3-address code, code generation for standard constructs with top-down and bottom-up parsers, translation of procedure calls, record structuring.

Code optimization: Loop optimization, DAG analysis, loop identification by flow dominance, depth-first search, reducible flow graphs, legal code motion, induction variables, data flow analysis, u-d and d-uchains, copy propagation, elimination of global sub-expressions, constant folding, code hoisting, forward and backward data flow equations, inter procedural data flow analysis.

Code generation: Problems in code generation, code generator, register assignment and allocation problems, usage count, code generation from DAG, peephole optimization.

Symbol table: Data structure and management, runtime storage administration, error detection and recovery; Lexical, syntactic and semantic errors, case studies with real life compilers.

References:
1. A. V. Aho, R. Sethi and J. Ullman: Compilers: Principles, Techniques and Tools, Addison-Wesley, California, 1986.
2. A. Appel: Modern Compiler Implementation in Java, Cambridge Univ. Press, London, 1997.

Operating Systems (CS 224)

Credits: 4

Total hours of instruction: 60

Evaluation: 70% Theory, 30% programming assignments

Introduction and history of Operating systems, structure and operations; processes and files;

Process management: interprocess communication, mutual exclusion, semaphores, wait and signal procedures, process scheduling and algorithms, critical sections, threads, multithreading;

Memory management: contiguous memory allocation, virtual memory, paging, page table structure, demand paging, page replacement policies, thrashing, segmentation, case study;

Deadlock: Shared resources, resource allocation and scheduling, resource graph models, deadlock detection, deadlock avoidance, deadlock prevention algorithms;

Device management : devices and their characteristics, device drivers, device handling, disk scheduling algorithms and policies

File management: file concept, types and structures, directory structure, cases studies, access methods and matrices, file security, user authentication;

Case Studies: UNIX and Linux operating systems;

References:
1. A. Silberschatz & P.B. Galvin, ‘Operating System concepts and principles’, Wiley India, 8th ed., 2009.
2. A. Tanenbaum, ‘Modern Operating Systems’, Prentice Hall India, 2003.
3. W. Stallings, ‘Operating Systems: Internals and design Principles’, Pearson Ed., LPE, 6th Ed., 2009.

4. M.J. Bach, ‘Design of Unix Operating system’, Prentice Hall, 1986.

Systems Programming (CS 228)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% programming assignments

1. Assembly language programming
   Endianness, SPIM assembly language for RISC based assembly programming, addressing modes, register files, callee-caller programming conventions,
1. Linux Virtual File System, and File I/O
   Inode and file strucutres, Programming interfaces for building a file system in Linux, blocking and non-blocking I/O, synchromized file I/O. Multiplexed file I/O: select/poll, scatter/gather, event poll, mmap.
2. Shell concepts
   job control, process grouping, command pipeline, file redirections, shell and environment variables.
3. Basic operating system functionality
   context switching, priority scheduling, process creation and overlay, signal management using siginfo_t, IPC, setuid concept,
4. Interfacing with the O.S.
   Talking to device files using ioctl, system calls for file operations.
5. Compiling, assembling, linking, loading
   Compilation and linking for Static, dynamic, and dynamic shared, position indepenent code, Importance of instruction selection in program translation,
6. Virtual Memory
   Address translation, Segmentation, Pagetables, TLB.
7. Device Drivers
   Kernel module programming, building a null driver, character and block device files, /proc file system
8. Concurrency basics
   Pthreads programming, fork and join, semaphores, mutual exclusion, the dining philosophers' problem.

References:
1. The Linux System programming, Robert Love, 2nd edition.
2. The Linux Kernel Module programming guide. Peter Jay Salzman, Michael Burian, Ori Pomerantz

Android Programming for Handheld Devices (CS 229)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Practical

1. Getting started - introduction to Android Studio; Android Project structure; App basics
2. Creating the First Basic App and using Emulator and device

3. Android MVC - using MVC for building app
4. Activities and Lifecycle
5. Communication among activities and Intents
6. Android UI Fragments
7. Android Lists and Adapters- RecyclerView,ListView, etc.
8. Android Layouts and Widgets
9. Advanced use of Fragments - Arguments, Nested Fragments
10. ViewPager
11. Dialogs and Fragments
12. Toolbar and Action Bars
13. SQLite Databases, Cursors and Asynctasks
14. Implicit Intents
15. Managing master-detail interfaces, Navigation Drawers
16. HTTP tasks
17. Background Tasks
18. Background Services
19. Maps
20. Material Design

References:

1. Instructor notes and exercises.
2. Google Android Training resources, https://developer.android.com/training/index.html

Computer Architecture (CS 234)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Introduction: Evolution of computer architecture, desired properties of the instruction set of a computer, instruction formats, addressing modes, architectural classifications based on multiplicity of data and instruction (SISD, SIMD, MISD and MIMD structures). CISC versus RISC architectures; Performance metric – different approaches.

Pipelining: Basic concepts, Performance of a static linear pipeline, instruction pipelining, hazards (structural, data and control hazards) and their remedies, instruction level parallelism (ILP). Super pipelining, super scalar processing, vector processing and pipelined vector processing.

Memory System: Memory hierarchy, Cache memory – fundamental concepts, reducing cache misses and cache miss penalty. Interleaved memory, virtual memory.

Interconnection Networks: Static vs. dynamic networks, desirable characteristics, example of popular static interconnection networks, dynamic networks – non-blocking, blocking, blocking rearrangable networks, unique full access multi-stage interconnection networks, examples.

Multiprocessors: Centralized shared-memory architectures, distributed shared-memory architectures, synchronization issues, models of memory consistency.

References:

1. D. A. Patterson and J. L. Hennessey: Computer Organization and Design: Hardware-Software Interface, Morgan Kaufmann, San Mateo, 1999.
2. J. L. Hennessey and D. A. Patterson: Computer Architecture: A Quantitative Approach, Morgan Kaufmann, San Mateo, 1999.
3. W. Stallings: Computer Organization and Architecture, Prentice Hall, New Jersey, 1999.
4. K. Hwang and F. A. Briggs: Computer Architecture and Parallel Processing, Tata McGraw Hill, New Delhi, 1984.

Database Management Systems (CS 253)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

1. Introduction and Conceptual Modeling
Database system concepts, three-schema Architecture, data independence, database administrator, database user, Client/Server Architecture, E-R diagram, mapping constraints, Keys, Generalization, Aggregation, Reducing E-R diagram to tables.

2. Relational Model: Concepts, constraints and Languages .
Structure of Relational database, Entity Integrity, Referential Integrity, Foreign Keys, Query languages, Relational algebra and relational calculus, SQL, views.

3. Database Design Theory and Methodology
Functional dependencies, Closure of a set of functional dependencies, Canonical cover, closure of attribute sets, Lossless decomposition, Dependency preservation, 1 NF, 2 NF, 3 NF, BCNF, Multivalued dependencies and 4 NF, Join dependencies and 5 NF.

4. Data Storage, Indexing and Query Processing
File organization, Sequential file, B+ tree index files, B-tree index file, Static hash Functions, Dynamic hash functions, Query processing and Query optimization.

5. Transaction Processing Concepts
Transaction, Properties of transaction, database recovery, shadow paging, recoverable schedule, serializable schedule; Concurrency control: Lock-Based protocol, Timestamp-Based protocol, Multiple granularity, Multiversion schemes; Deadlock Handling.

6. Database Security
Discretionary access control, Mandatory access control and multi-level security, statistical database security, Introduction to flow control, Encription and public key infrastructures, privacy issues and preservation.

References:

1. H. F. Korth and A. Silberschatz: Database System Concepts, McGraw Hill, New Delhi, 1997.
2. R. A. Elmasri and S. B. Navathe: Fundamentals of Database Systems, 3rd ed., Addison-Wesley, 1998.
3. R. Ramakrishnan: Database Management Systems, 2nd ed., McGraw Hill, New York, 1999.
4. C. J. Date, A. Kannan and S. Swamynathan, *An Introduction to Database Systems*, Pearson Education, Eighth Edition, 2009.

5. J D Ullman : Principles of Database Systems, Computer Science Press; 2nd edition (December 1982)

Theory of NP-Completeness (CS 300)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Decision problems, languages and encoding systems
Deterministic Turing Machines and the class P
Nondeterministic Turing Machines and the class NP
Relationship between P and NP
Polynomial reduction and NP-completeness
Cook-Levin's theorem

Proving NP-completeness results
Basic NP-complete problems, 3SAT, 3DM, VC,CLIQUE, IS, HC, PARTITION
Techniques for proving NP-completeness, restriction, local replacement
and component design

Using NP-completeness to analyze problems, number problems and strong
NP-completeness

NP-hardness, turing reducibility and NP-hard problems

Coping with NP-complete problems, performace guaruntees for approximation
algorithms
Applying NP-completeness to approximation problems

Polynomial hierarchy, classes PSPACE, L, NL, EXP and NEXP

References:

1. M. R. Garey and D. S. Johnson, Computers and Intractibility: A guide to the theory of NP-completeness, W. H. Freeman, 1979.
2. S. Arora and B. Barak. Complexity Theory: A Modern Approach . Cambridge University Press, 2009
3. J.E. Hopcroft and J.D. Ullman. Introduction to Automata Theory, Languages and Computation . Addison-Wesley, 1979.
4. M. Sipser. Introduction to the Theory of Computation . PWS Publishing Company, 1997.
5. O. Goldreich. Computational Complexity: A Conceptual Perspective , Cambridge University Press, 2008.
6. Jon Kleinberg, Eva Tardos; Algorithm Design, Pearson education, 2008
7. T. H. Cormen, C.E. Leiserson and R.L.Rivest: Introduction to Algorithms, Prentice Hall of India, New Delhi, 2009 (3rd Ed)

Approximation and Onilne Algorithms (CS 312)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Approximation Algorithm:
Performance Measure, Greedy Algorithm, Unweighted Vertex Cover Problem
Minumum-Degree Spanning Tree, Minimum Weight Spanning Tree, The Traveling-Salesman Problem,The k-Center Problem, Multiway Cut and K-Cut Problems, Scheduling Jobs with Deadlines on a Single Machine, Scheduling Jobs on Identical Parallel Machines, The Set Cover Problem, An Application of Set Cover to Art Gallery problems, Shortest Superstring Problem Rounding Data and Dynamic Programming, The Knapsack Problem, The Bin-Packing Problem, The Primal-Dual Method, Weighted Vertex Cover Problem

Online Algorithms:
Competitive Analysis, The Paging Problem, Amortized Analysis, List Update Problem, Scheduling Jobs on Identical Parallel Machines, Graph Colouring, Machine Learning, K-Server Problem, Target Searching in an Unbounded Region and Target Searching in Streets

References:

1. M. R. Garey and D. S. Johnson, Computers and Intractibility: A guide to the theory of NP-completeness, W. H. Freeman, 1979.
2. R. Motwani, Lecture Notes on Approximation Algorithms, Volume 1, No. STAN-CS-92-1435, Stanford University, 1992.
3. D. P. Williamson and D. B. Shmoys, The Design of Approximation Algorithms, Cambridge University Press, 2011.
4. Vijay Vazirani, Approximation algorithms, Springer-Verlag, 2001.
5. S. Albers, Competitive Online Algorithms, Lecture notes, Max Plank Institute, Saarbrucken, 1996.
6. S. K. Ghosh and R. Klein, Online algorithms for searching and exploration in the plane, Computer Science Review, vol. 4, pp. 189-201, 2010.

Distributed Computing Systems (CS 321)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 20% Lab assignments.

Introduction to Distributed System: Introduction, Examples of distributed system, Resource sharing, Challenges

Operating System Structures: Review of structures: monolithic kernel, layered systems, virtual machines. Process based models and client server architecture; The micro-kernel based client-server approach.

Communication: Inter-process communication , Remote Procedure Call, Remote Object Invocation, Tasks and Threads.
Examples from LINUX

Theoretical Foundations: Introduction. Inherent Limitations of distributed Systems. Lamport's Logical clock. Global State Distributed

Mutual Exclusion: Classification of distributed mutual exclusion algorithm. NonToken based Algorithm:Lamport's algorithm,
Ricart-Agrawala algorithm. Token based Algorithm: Suzuki-Kasami's broadcast algorithm.

Distributed Deadlock Detection: Deadlock handling strategies in distributed systems. Control organizations for distributed deadlock detection. Centralized and Distributed deadlock detection algorithms: Completely Centralized algorithms, path pushing, edge chasing, global state detection algorithm.

Distributed file systems: Issues in the design of distributed file systems: naming, transparency, update semantics and fault resilience. Use of the Virtual File System layer. Examples of distributed systems including Sun NFS, the Andrew filestore, CODA file system and OSF DCE.

Protection and Security: Requirements for protection and security regimes. The access matrix model of protection. System and user modes, rings of protection, access lists, capabilities. User authentication, passwords and signatures. Use of single key and public key encryption.

Distributed Shared Memory: Architecture and motivations. Algorithms for implementing DSM. Memory Coherence

Case Study: Kerberos, CORBA

References:

1. Distributed systems : principles and paradigms by Andrew S. Tanenbaum; Maarten Van Steen, Prentice Hall of India, New Delhi 2002
2. Distributed computing : principles, algorithms, and systems by Ajay D. Kshemkalyani; Mukesh Singha, Cambridge University Press, Cambridge, New York 2008
3. Distributed systems: concepts and systems by George Coulouris; Jean Dollimore; Tim Kindberg, Pearson Education Inc. C2005
4. Distributed operating systems by Andrew S. Tanenbaum, Publisher: Pearson Education c1995
5. Advanced concepts in operating systems : distributed, database, and multiprocessor operating systems by Mukesh Singhal; Niranjan G. Shivaratri, McGraw Hill Education (india) 2001
6. Distributed systems architecture: a middleware approach by Arno Puder; Kay Romer; Frank Pilhofer, Morgan Kaufmann, Elsevier c2006
7. Distributed systems : an algorithmic approach by Sukumar Ghosh, CRC Press, Taylor & Francis Group 2014
8. Distributed operating systems : concepts and design by Pradeep K. Sinha, PHI Learning 2014
9. Distributed operating systems and algorithms by Randy Chow; Theodore Johnson, Pearson India 2009
10. Distributed Algorithms by Nancy A. Lynch, Morgan Kaufmann Publishers 2005
11. Introduction to distributed algorithms by Gerard Tel, Cambridge University Press 2004
12. Distributed algorithms: an intuitive approach by Wan Fokkink, The MIT Press c2013

Discrete Event Systems (CS 323)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Systems and models: system and control basics, concept of system, concept of state, state modelling, concept of control and feedback, discrete time systems, d

Discrete event systems, concept of event system classification

Languages and automata

Supervisory control, feedback control with supervisors,
specifications on controlled system, partial controllability,

Nonblocking control

Control with modular specifications, control under partial observation

Decentralized control, conjunctive, disjunctive and combined architecty=ure,
decentralized supervisors, undecidability in decentralized supervisors

Petri nets, basics, analysis and control of Petri nets

Timed and hybrid models, timed automata, timed Petri nets, dioid algebras,
timed automata with guards, hybrid automata

Stochastic timed automata, introduction

Markov chains, discrete-time markov chains, continuous time markov chains

Introduction to queueing theory

References:

1. Introduction to Discrete Event Systems - Second Edition by C. Cassandras and S. Lafortune, Springer, 2008
2. Synchroniyation and Linearity: An Algebra for Discrete Event Systems by F. Baccelli, G. Cohen, G.J. Olsder, J.P. Quadrat, MIT Press, 1995
3. Discrete-event System Simulation by Jerry Banks, John S. Carson, Barry L. Nelson Prentice Hall, 2013 (5ed)
4. Discrete Event Systems: Diagnosis and Diagnosability by Moamar Sayed-Mouchaweh, Springer New York, 2014
5. Modeling and Simulation of Discrete Event Systems by Byoung Kyu Choi, DongHun Kang, John Wiley & Sons, 2013

Machine Learning (CS 340 / DA220)

Credits: 4

Total hours of instruction: 60

Evaluation: 50% Theory , 50% Practical

1. Overview of machine learning: idea of supervised and unsupervised learning, regression vs classification, concept of training and test set, classification vs clustering and significance of feature engineering
2. Linear regression: least square and least mean square methods
3. Bayes decision rule: bayes theorem, bayes classifier, minimum distance classifier, linear discriminant function
4. Naive bayes classifier: gaussian model, multinomial model, bernoulli model
5. k-Nearest Neighbor (kNN) decision rule: idea of kNN classifier, distance weighted kNN decision rule and other variations of kNN decision rule
6. Perceptron learning algorithm: incremental and batch version, proof of convergence, XOR

problem, two layer perceptrons to resolve XOR problem, introduction to multi-layer perceptrons

7. Logistic regression and maximum margin classifier
8. Support vector machine (SVM): hard margin and soft margin SVM classifier
9. Cross validation, parameter tuning and evaluation techniques for the classifiers
10. Text classification (case study for data classification 1): overview of text data, stemming and stopword removal, tf-idf weighting scheme and n-gram approach, performance of different classifiers to classify a newswire e.g., Reuters-21578.
11. Data clustering: overview, cluster validity index
12. Partitional clustering methods: k-means, bisecting k-means, k-medoid, buckshot
13. Hierarchical clustering techniques: single linkage, average linkage and group average hierarchical clustering algorithms, density based clustering technique e.g., DBSCAN
14. Feature engineering: overview of feature selection, supervised and unsupervised feature selection techniques, feature extraction techniques e.g., principal component analysis
15. Sentiment analysis (case study for data clustering 1): overview, description of a data set of interest for sentiment identification, Wordnet (English Lexical Database), sentiment analysis using Wordnet

References:

1. Introduction to Machine Learning by E. Alpaydin.
2. The Elements of Statistical Learning by J. H. Friedman, R. Tibshirani, and T. Hastie.
3. Pattern Recognition by S. Theodoridis and K. Koutroumbas.
4. Pattern Classification by R. O. Duda, P. E. Hart and D. G. Stork.
5. Pattern Recognition and Machine Learning by C. M. Bishop.
6. Machine Learning by T. Mitchell
7. Introduction to Information Retrieval by C. D. Manning, P. Raghavan and H Schutze

Graph Theory (CS 302)

1. Introduction to Graphs & its Applications, Basics of Paths, Cycles, and Trails, Connection, Bipartite Graphs, Eulerian Circuits, Vertex Degrees and Counting, Degree-sum formula, The Chinese Postman Problem and Graphic Sequences.
2. Trees and Distance, Properties of Trees, Spanning Trees and Enumeration, Matrix-tree computation, Cayley's Formula, Prufer code
3. Matchings and Covers, Hall's Condition, Min-Max Theorem, Independent Sets, Covers and Maximum Bipartite Matching, Augmenting Path Algorithm, Weighted Bipartite Matching, Hungarian Algorithm.
4. Stable Matchings and Faster Bipartite Matching, Factors & Perfect Matching in General Graphs, Matching in General Graphs: Edmonds' Blossom Algorithm
5. Connectivity and Paths: Cuts and Connectivity, k-Connected Graphs, Network Flow Ford-Fulkerson Labeling Algorithm, Max-Flow Min-cut Theorem, Menger's Proof using Max-Flow Min-Cut Theorem.
6. Vertex Coloring and Upper Bounds, Brooks' Theorem and Color-Critical Graphs, Counting Proper Colorings.
7. Planar Graphs, Characterization of Planar Graphs, Kuratowski's Theorem, Wagner's Theorem.
8. Line Graphs and Edge-coloring, Hamiltonian Graph, Traveling Salesman Problem and NP-Completeness, Dominating Sets.

References:

1. D.B. West, Introduction to Graph Theory, Prentice Hall, 2001
2. Jon Kleinberg and Eva Tardos, Algorithm Design, Addison-Wesley, 2005
3. J.A.Bondy and U.S.R.Murty: Graph Theory, Springer, 2008.
4. R.Diestel: Graph Theory, Springer(low price edition) 2000.
5. F.Harary: Graph Theory, Narosa, (1988)

6. C. Berge: Graphs and Hypergraphs, North Holland/Elsevier, (1973)

Introduction to Discrete Optimization (CS 222)

Credits: 4

Total hours of instruction: 60

Evaluation: 80% Theory, 30% practical

1. Linear Programming: A brief review of simplex and revised simplex algorithms, Bland's rule, duality theory, large scale linear programmes, computational complexity of simplex method, polynomial time algorithms – ellipsoidal and Karmarkar's methods.

2. Integer Programming: All integer and mixed integer programming problems, cutting planes and branch and bound algorithms, introduction to the ideas of NP-completeness, travelling salesman and other related problems.

3. Non-linear Programming: General constrained mathematical programming problems, Kuhn-Tucker-Lagrangian necessary and sufficient conditions, interior point methods, standard algorithms like feasible direction and gradient projections convergence of the methods, convex programming problems, quadratic programming.

References:

1. R. J. Vanderbei: Linear Programming Foundations and Extensions, Kluwer Academic Publishers, Boston/London, 1997.
2. D. G. Luenberger: Introduction to Linear and Non-Linear Programming, Addison-Wesley Publishing Co., London/Amsterdam,1984.
3. C. H. Papadimitriou and K. Steiglitz: Combinational Optimization, Prentice Hall, Englewood Cliffs, 1982.
4. R. Garfinkel and G. Nemhauser: Integer Programming, John Wiley, New York, 1976.
5. G. Nemhauser and L. Wolsey: Integer and Combinational Optimization, Wiley, New York, 1988.
6. D. Bertsekas: Non-Linear Programming. Athena Scientific, Belmont, Mass., 1995.
7. S. Nash and A. Sofer: Linear and Non-Linear Programming, McGraw Hill, New York, 1996.
8. F. Hillier and G. Liebermann: Introduction to Mathematical Programming, McGraw Hill, 1995.
9. K. G. Murty: Linear and Combinatorial Programming, John Wiley, New York, 1976.
10. M. Bazaraa, J. Jarvis and H. Sherali: Linear Programming and Network Flows, Wiley, New York, 1977.
11. W. I. Zangwill: Non-Linear Programming, Prentice Hall, New Jersey, 1969.
12. R. Fletcher: Practical Methods of Constrained Optimization, John Wiley, Chichester, 1981.

Computational Complexity (CS 301)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Introduction; P and NP:
Review of Turing machines, universal Turing machines, and uncomputable functions

P, NP, coNP, and NP-Completeness:

P vs. NP; NP vs. coNP; and NP-completeness, NP-completeness of SAT and other problems. Search vs. decision and self-reducibility.

Diagonalization:
Time/space hierarchy theorems, Ladner's theorem. Relativization of the P vs. NP question.

Space complexity:
PSPACE and PSPACE-completeness; NL and NL-completeness., Savitch's theorem; the Immerman-Szelepcsenyi theorem.

The polynomial hierarchy:
The polynomial hierarchy. Time-space tradeoffs for SAT, PH in terms of oracle machines.

Non-uniform complexity:
Introduction to non-uniform complexity., Circuit complexity and P/poly. The Karp-Lipton theorem. Logarithmic-depth circuits., Kannan's circuit lower bound. Logarithmic-depth circuits.

Randomized computation:
RP, BPP, ZPP. Error reduction. Probabilistic algorithms, Relation of BPP to the polynomial hierarchy and non-uniform computation. BPP-completeness and promise problems.

Randomized space complexity:
RL and BPL. Random walks on undirected graphs, Markov chains, and random walks on undirected graphs

Interactive proofs:
Interactive proofs, MA, and AM., Graph non-isomorphism in AM, An interactive proof for coNP, IP=PSPACE. Introduction to zero-knowledge proofs.

Zero-knowledge proofs:
Honest-verifier and dishonest-verifier perfect zero-knowledge proofs. Computational zero-knowledge proofs; perfect zero-knowledge arguments.

The PCP theorem:
The PCP theorem and applications to proving inapproximability, A proof that NP is in PCP

The PCP theorem; the complexity of counting:
A proof that NP is in PCP(poly, 1), part 2. #P and #P-completeness.

The complexity of counting:
Hardness of unique-SAT. Approximating #P with an NP oracle. Toda's theorem.

Time-bounded derandomization:
Finish Toda's theorem. Complexity-theoretic pseudorandom generators. The Nisan-Wigderson construction.

Space-bounded derandomization, error reduction:
Unconditional derandomization for space-bounded algorithms using Nisan's PRG. Error reduction using Nisan's PRG.

Circuit lower bounds:
Parity is not in AC0. Natural proofs.

References:

1. Computational Complexity - A Modern Approach by Sanjeev Arora and Boaz Barak
2. Computational Complexity Theory by Steven Rudich and Avi Wigderson (Editors)
3. Gems of Theoretical Computer Science by Schoening and Pruim
4. The Nature of Computation by Moore and Mertens
5. D.Z. Du and K.I. Ko : Theory of Computational Complexity.
6. The Complexity Theory Companion by Hemaspaandra and Ogihara
7. C.H. Papadimitriou : Computational Complexity.

Pattern Recognition (CS 245)

Credits: 4

Total hours of instruction: 60

Evaluation: 50% Theory, 50% Practical

1. Introduction: what is pattern recognition, applications and scopes, idea of supervised and un-supervised learning, regression vs classification, concept of training and test set, classification vs clustering and significance of feature engineering
2. Linear regression: least square and least mean square methods
3. Bayes decision rule: bayes theorem, bayes classifier, minimum distance classifier
4. Linear discriminant function for data analysis
5. Perceptron learning algorithm: incremental and batch version, proof of convergence, XOR problem, two layer perceptrons to resolve XOR problem, introduction to multi-layer perceptrons
6. Support vector machine (SVM): hard margin and soft margin SVM classifier
7. k-Nearest Neighbor (kNN) decision rule: idea of kNN classifier, distance weighted kNN decision rule and other variations of kNN decision rule
8. Cross validation, parameter tuning and evaluation techniques for the classifiers
9. Text classification (case study for data classification 1): overview of text data, stemming and stopword removal, tf-idf weighting scheme and n-gram approach, performance of different classifiers to classify a newswire e.g., Reuters-21578.
10. Data clustering: overview, different distance functions and similarity measures, cluster validity index
11. Partitional clustering methods: k-means, bisecting k-means, k-medoid, buckshot
12. Hierarchical clustering techniques: single linkage, average linkage and group average hierarchical clustering algorithms, density based clustering technique e.g., DBSCAN
13. Expectation maximization method
14. Feature engineering: overview of feature selection, supervised and unsupervised feature selection techniques, feature extraction techniques - principal component analysis (PCA) and kernel PCA
15. Sentiment analysis (case study for data clustering 1): overview, description of a data set of interest for sentiment identification, Wordnet (English Lexical Database), sentiment analysis using Wordnet

References:

1. Pattern Recognition by S. Theodoridis and K. Koutroumbas.
2. Pattern Classification by R. O. Duda, P. E. Hart and D. G. Stork.
3. The Elements of Statistical Learning by J. H. Friedman, R. Tibshirani, and T. Hastie.
4. Pattern Recognition and Machine Learning by C. M. Bishop.

Discrete Mathematics (CS 207)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Combinatorics: Multinomial theorem, principle of inclusion exclusion; Recurrence relations – classification, summation method, extension to asymptotic solutions from solutions for subsequences; Linear homogeneous relations, characteristic root method, general solution for distinct and repeated roots, non-homogeneous relations and examples, generating functions and their application to linear homogeneous recurrence relations, non-linear recurrence relations, exponential generating functions, brief introduction to Polya theory of counting.

Graph Theory: Graphs and digraphs, complement, isomorphism, connectedness and reachability, adjacency

matrix, Eulerian paths and circuits in graphs and digraphs, Hamiltonian paths and circuits in graphs and tournaments, trees; Minimum spanning tree, rooted trees and binary trees, planar graphs, Euler's formula, statement of Kuratowskey's theorem, dual of a planer graph, independence number and clique number, chromatic number, statement of Four-color theorem, dominating sets and covering sets.

Logic: Propositional calculus – propositions and connectives, syntax; Semantics – truth assignments and truth tables, validity and satisfiability, tautology; Adequate set of connectives; Equivalence and normal forms; Compactness and resolution; Formal reducibility – natural deduction system and axiom system; Soundness and completeness.

Introduction to Predicate Calculus: Syntax of first order language; Semantics – structures and interpretation; Formal deductibility; First order theory, models of a first order theory (definition only), validity, soundness, completeness, compactness (statement only), outline of resolution principle.

References:

1. J. L. Mott, A. Kandel and T. P. Baker: Discrete Mathematics for Computer Scientists, Reston, Virginia, 1983.
2. D. F. Stanat and D. E. McAllister: Discrete Mathematics in Computer Science, Prentice Hall, Englewood Cliffs, 1977.
3. C. L. Liu: Elements of Discrete Mathematics, 2nd ed., McGraw Hill, New Delhi, 1985.
4. R. A. Brualdi: Introductory Combinatorics, North-Holland, New York, 1977.
5. Reingold et al.: Combinatorial Algorithms: Theory and Practice, Prentice Hall, Englewood Cliffs, 1977.
6. J. A. Bondy and U. S. R. Murty: Graph Theory with Applications, Macmillan Press, London, 1976.
7. N. Deo: Graph Theory with Applications to Engineering and Computer Science, Prentice Hall, Englewood Cliffs, 1974.
8. E. Mendelsohn: Introduction to Mathematical Logic, 2nd ed. Van-Nostrand, London, 1979.
9. L. Zhongwan: Mathematical Logic for Computer Science, World Scientific, Singapore, 1989.
10. F. S. Roberts: Applied Combinatorics, Prentice Hall, Englewood Cliffs, NJ, 1984.
11. Lewis and Papadimitriou: Elements of Theory of Computation (relevant chapter on Logic), Prentice Hall, New Jersey, 1981.

Linear Algebra (M204a)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

Linear equations: Systems of linear equations, Matrices and elementary row operations, Row reduced Echelon matrices, Matrix multiplication, Invertible matrices, Transpose of a matrix, Systems of homogeneous equations, Equivalence of row rank and column rank of a matrix, Determinant and volume of the fundamental parallelepiped, Permutation matrices, Cramer’s rule

Vector spaces: Vector spaces and subspaces, Bases and dimensions, Coordinates and change of bases, Direct sums

Linear transformations: The Rank-Nullity theorem, Matrix of a linear transformation, Linear operators and isomorphism of vector spaces, Determinant of a linear operator, Linear functionals, Annihilators, The double dual

Eigenvalues and eigenvectors: Eigenvalues and eigenvectors of matrices, The characteristic polynomial, Algebraic and geometric multiplicities of eigenvalues, Diagonalizability, Cayley-Hamilton theorem, Solving linear recurrences

Bilinear forms: Matrix of a bilinear form, Symmetric and positive definite bilinear forms, Normed spaces, Cauchy-Schwarz inequality and triangle inequality, Angle between two vectors, Orthogonal complement, Projection, Gram-Schmidt orthogonalization, Hermitian operators, The Spectral theorem

Introduction to linear programming: Bounded and unbounded sets, Convex functions, Convex cone, Interior points and boundary points, Extreme points or vertices, Convex hulls and convex polyhedra, Supporting and separating hyperplanes, Formulating linear programming problems, Feasible solutions and optimal solutions, Graphical method, The basic principle of Simplex method

References:

1. M. Artin, Algebra, Prentice Hall.
2. K. M. Hoffmann, R. Kunze, Linear Algebra, Prentice Hall.
3. G. Strang, Introduction to Linear Algebra, Wellesley-Cambridge Press.
4. L. I. Gass, Linear Programming, Tata McGraw Hills.
5. G. Hadley, Linear Programming, Narosa Publishing House.

Data Mining (CS 351)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

(a) Introduction: Introduction to data mining and knowledge discovery from databases. Scalability issues of data mining algorithms.

Introduction to Data warehousing: General principles, modelling, design, implementation, and optimization. Data preparation:

Preprocessing, sub-sampling, feature selection. Classification and prediction: Bayes learning, discriminant analysis, decision trees,

CART, C4.5 etc, neural learning, support vector machines, active learning. Combination of classifiers/ ensemble learning. Associations,

dependence analysis, correlation, rule generation— a priori algorithm, FP Trees etc. and evaluation. Cluster analysis and deviation

detection: Partitioning algorithms, density based algorithms, hierarchical algorithms, model based algorithms, grid based algorithms,

graph theoretic clustering etc. Temporal and spatial data mining: Mining complex types of data. Visualization of data mining results.

Advanced topics: High performance computing for data mining, distributed data mining, soft computing tools for data mining.

Applications of data mining in bioinformatics, information retrieval, web mining, image and text mining.

(b) Four lectures per week

(c) Theory 70% and Assignment 30%

(d) References:

1. J. Han, M. Kamber: Data Mining: Concepts and Techniques, Morgan Kaufmann, 2000 2. D. J. Hand, H. Mannila and

P. Smyth: Principles of Data Mining, MIT Press, 2000 3. M. Berry and G. Linoff: Mastering Data Mining, John Wiley & Sons, 2000. 4. A.

K. Pujari: Data Mining Techniques, Sangam Books Ltd., 2001.

Algorithms in Mobile Computing (CS 322)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

(a) Introduction: Challenges in mobile computing, coping with uncertainties, resource scarcity, bandwidth, energy etc. Cellular architecture, co-channel interference, frequency reuse, capacity increase by cell splitting. Evolution of mobile system: CDMA, FDMA,

TDMA, GSM. Mobility Management: Handoff, types of handoffs; location management, HLR-VLR scheme, hierarchical scheme, predictive location management schemes. Mobile IP, cellular IP. Publishing and Accessing Data in Air: Pull and push based data delivery models, data dissemination by broadcast, broadcast disks, directory service in air, energy efficient indexing scheme for push based data delivery. File System Support for Mobility: Distributed file sharing for mobility support, storage manager for mobility support.

Ad hoc and Sensor Networks: Routing Algorithms and Protocols, load-balancing, scheduling for reduced energy, coverage and connectivity problems. Mobile Transaction and Commerce: Models for mobile transaction. Kangaroo and joey transactions, team transaction. Recovery model for mobile transactions. Electronic payment and protocols for mobile commerce.

(b) Operating systems, Distributed Systems, Computer Architecture and Computer Networks.

(c) Three lectures and 1 tutorial

(d) Theory 70% and Assignment 30%

(e) References:

T. Imielinski and H. F. Korth: Mobile Computing, Kluwer Academic Publishers, Boston, 1996. 2. A. A. Helal, B.

Haskell, J. L. Carter, R. Brice, D. Woelk and M. Rusinkiewicz: Any Time, Anywhere Computing: Mobile Computing Concepts and

Technology, Kluwer International Series in Engineering and Computer Science, 1999. 53 3. C. Perkins, S. R. Alpert and B. Woolf:

Mobile IP: Design Principles and Practices, Addison Wesley Longman, 1997. 4. Y. Lin and I. Chlamtac: Wireless and Mobile

Network Architecture, Wiley India Pvt. Ltd., New Delhi, 2001.

Web Technologies (CS251)

Credits: 4

Total hours of instruction: 60

Evaluation: 100% Theory

An introductory chapter on basic Web Architecture like 2-tier, 3tier, N tier, MVC

Http, Https and extended valuation, WWW, FTP,. Case study for www., DNS, domains and sub-domains.

A brief overview of Server architecture of a modern Web Server.

HTML5 with CSS3, XML, Java Script.

Static vs Dynamic web pages, Handing various media like images, videos, audio, streaming, AVplayers,

PHP: understanding PHP.ini file on the server, PHP Packages and compilation. Study one package in some detail.

Advanced Java that includes Applet and Servlet, Java RMI and Socket Programming should be introduced.

Web Services, Cloud computing, cloud storage, Cloud services, Cloudflair

Performance - Concerns regarding performance at server side, middleware and client end.,

Simple concepts illustrating the need for performance tuning, popular practices used for improving performance.

Web Security: Why security, illustrate a simple security loophole, List of standard security issues and how to tackle them.

Reference:

Internet resources