M.Tech in Computing and Mathematics
Syllabi of Core Courses and a Representative Set of Electives
(Approved Version as on 24.06.2020)

Course Code and Title	**CS5009: Algorithms**				
Programme	**M.Tech in Computing and Mathematics**	**Year of study**	**1**	**Semester**	**1**
Course credit	**3-1-0-4**				
Course category	**PMT**				
Prerequisite, if any	**-**				

Course Objective

The objective of the course is to help students learn basic algorithms and their rigorous analysis. The analysis will involve proofs of correctness and efficiency analysis of implementation using appropriate data structures. The course also will provide a detailed introduction to the different algorithm design paradigms for problems from various domains.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Preliminaries :** Asymptotic notations, Efficiency of algorithms, Notions of time and space complexity.	3	1	0
2	**Design Paradigms:** Greedy Strategy, Divide & Conquer and Dynamic Programming with applications in different problem domains like sorting, searching, string matching, scheduling and simulation.	15	5	0
3	**Graph Algorithms :** Depth First Search, Breadth First Search, Minimum Spanning Tree algorithms, Shortest path algorithms, Network flow algorithms.	24	8	0
	Total	42	14	0

Learning Outcomes

1. To acquire basic mathematical tools and techniques for algorithm design and analysis.
2. To familiarize with basic data structures and develop the ability to choose the appropriate data structure for designing efficient algorithms.
3. To develop the ability to analyze the running time and prove the correctness of algorithms.
4. To be able to design efficient algorithms for computational problems using various algorithm design techniques taught in the course.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Introduction to Algorithms (3rd Edition) by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. PHI Learning, 2009. ISBN:978-81-203-4007-7.

2. Algorithms by Sanjoy Dasgupta, Christos H. Papadimitriou, Umesh Vazirani. McGraw-Hill Education, 2006. ISBN: 978-0073523408.

References:

1. Algorithm Design by Jon Kleinberg and Eva Tardos. Pearson, 2015. ISBN:978-93-325-1864.
2. The Design and Analysis of Algorithms by Dexter C. Kozen, Springer, 1992. ISBN: 978-0-387-97687-7.

Course Code and Title	CS5013 Topics in Discrete Mathematics				
Programme	M.Tech in Computing and Mathematics	Year of study	1	Semester	1
Course credit	3-0-0-3				
Course category	PMT				
Prerequisite, if any	None				

Course Objective

This is a course intended to prepare starting graduate students with a selected set of topics from Discrete Mathematics which play a major role in Theoretical Computer Science. The emphasis will be on laying these foundations as rigorously as possible.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Naive Set Theory**: Cardinality of sets, Schröder–Bernstein Theorem. Uncountable sets. Diagonalisation, Cantor’s Theorem.	10	0	0
2	**Number Theory**: Induction, and Well Ordering Principle. Proofs using Induction and Well Ordering Principle. Bézout's identity. Euclid’s Algorithm, Structural Induction. Fermat’s Little Theorem, Euler’s theorem, Modular Arithmetic. Chinese Remainder Theorem.	16	0	0
3	**Introduction to First Order Logic:** Syntax of First Order Logic, Scope and Binding. Semantics. Models, First Order Calculus, Soundness and Completeness of First Order Calculus, Compactness and its Applications.	16	0	0
	Total	42	0	0

Learning Outcomes: After successful completion of this course, a student will be able to

1. Follow a rigorous mathematical proof and identify gaps in reasoning if any.
2. Reason about sizes of infinite sets.
3. Apply standard proof techniques to prove statements about simple mathematical structures.
4. Prove properties of recursively defined structures using structural induction.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Discrete Mathematics (2nd edition) by Norman L. Biggs, OUP, 2002, ISBN-13: 978-0198507178.
2. Part II (Logic) of the Computational Complexity book by Christos H. Papadimitrou, Pearson, 1993, ISBN-13: 9780201530827.

References:

1. Discrete Mathematics: Elementary and Beyond by László Lovász, József Pelikán, Katalin Vesztergombi, Springer 2003, ISBN-13: 978-0387955858.
2. Discrete Mathematics and Applications by Kenneth Rosen, 7th Edition, McGraw-Hill Education 2012, ISBN-13: 978-0073383095.

Course Code and Title	**CS5017 Theory of Computation**				
Programme	**M.Tech in Computing and Mathematics**	**Year of study**	**1**	**Semester**	**1**
Course credit	**3-1-0-4**				
Course category	**PMT**				
Prerequisite, if any					

Course Objectives: The course will provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical view towards algorithmic design and in general computation itself.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Regular Languages & Finite Automata**: Regular Languages and Regular Expressions, Deterministic and Non-deterministic Finite Automata, Kleene's Theorem, Pumping Lemma, Myhill-Nerode Theorem.	12	4	0
2	**Introduction to Context-free Languages & Pushdown Automata**: Context-free Languages and Grammars, Ambiguity, Chomsky Normal Form, CYK Algorithm, Pumping Lemma, Introduction to Deterministic and Nondeterministic Pushdown Automata.	9	3	0
3	**Turing Machines & Recursive Languages**: Mathematical modelling of computation, Deterministic Turing Machines, Church-Turing Thesis, Chomsky Hierarchy, Universal Turing Machines. Recursive and Recursively Enumerable Languages. Non-recursive Languages and Undecidable Problems, the Halting Problem. Reduction	12	4	0
4	**Complexity:** Resource-bounded computation, Classes P and NP, Polynomial time reductions, NP-completeness.	9	3	0
	Total	42	14	0

Learning Outcomes: After completion of this course, a student will be able to

1. Give the mathematical definition of various computational models and state and prove their limitations.
2. To explain important notions in computing like nondeterminism, reductions and resource boundedness.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Introduction to Languages and The Theory of Computation (4th Edition) by John C. Martin, McGraw-Hill Publishers, 2011. ISBN: 9780073191461.
2. Automata and Computability by Dexter C. Kozen. Springer Publishers 2007. ISBN: 0387949070.

References:

1. Elements of Computation Theory by Arindama Singh, Springer-Verlag London, 2009. ISBN: 978-1-4471-6142-4.
2. Introduction to Automata Theory, Languages and Computation by Hopcroft, Motwani, and Ullman. 3rd Edition, Pearson Publishers, 2006. ISBN:0321462254.
3. Elements of the Theory of Computation by H. R. Lewis and C.H. Papadimitriou, Prentice Hall Publishers, 1981. ISBN-13: 978-0132624787.

Course Code and Title	CS5010 Graph Theory and Combinatorics				
Programme	M.Tech in Computing and Mathematics	Year of study	1	Semester	2
Course credit	3-0-0-3				
Course category	PMT				
Prerequisite, if any					

Course Objective
The objective of this course is to introduce the students to various techniques in graph theory and combinatorics.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	Matching, Hall's Theorem, Kőnig's Theorem, Connectivity, Coloring, Planar Graphs, Eulerian Graphs, Hamiltonian cycles, Dirac's Theorem, Ore's Theorem.	12	0	0
2	Introduction to Ramsey Theory and Extremal Graph Theory.	8	0	0
3	Ordinary, Exponential and other special Generating Functions, Formal Power Series.	10	0	0
4	Linear Algebraic Arguments like Dimensionality, Orthogonality and Rank arguments. Introduction to Spectral Graph Theory.	12	0	0
	Total	42	0	0

Learning Outcomes: By the end of this course, students will be able to state, explain, prove, and apply fundamental results in graph theory and combinatorics. They will be able to analyse these proofs and apply these proof techniques to solve similar problems.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Reinhard Diestel, Graph Theory, Fifth Edition, Springer-Verlag Berlin Heidelberg, 2017, ISBN: 978-3-662-53621-6.
2. F. R. K Chung, Spectral Graph Theory, ISBN: 978-0-8218-0315-8.
3. László Babai and Péter Frankl, Linear Algebra Methods in Combinatorics With Applications to Geometry and Computer Science, Preliminary Version 2, 1992.

References:

1. Adrian Bondy and U. S. R. Murty, Graph Theory, Springer, 2008 Edition, ISBN: 978-1849966900.
2. Chris Godsil and Gordon Royle, Algebraic Graph Theory, Springer, 2001, ISBN: 978-0-387-95220-8.

3. Norman Biggs, Algebraic Graph Theory, Cambridge University Press, 1994. ISBN-13: 978-0521458979.
4. D.B West, Introduction to Graph Theory, Prentice Hall, 1996. ISBN: 9780132278287..

Course Code and Title	CS5014: Foundations of Data Science and Machine Learning				
Programme	M.Tech in Computing and Mathematics	**Year of study**	1	**Semester**	2
Course credit	3-0-0-3				
Course category	PMT				
Prerequisite, if any	Probability and Linear Algebra				

Course Objectives: While traditional areas of computer science remain highly important, increasingly researchers of the future will be involved with using computers to understand and extract usable information from massive data arising in applications. The main objective of this course is to introduce students to the theoretical and mathematical foundations of data science. This course will be rigorous, and will explore the rich and fascinating math behind some of the popular techniques and intellectual ideas of modern day data science and machine learning.

Course Content

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**High-dimensional space:** law of large numbers, the geometry of high dimensions	6	0	0
2	**Best-Fit subspaces and SVD**: Introduction, singular vectors, singular value decomposition (SVD), best k-rank approximations, left singular vectors, eigenvectors, applications of SVD	9	0	0
3	**Random walks and Markov Chains**: Introduction, stationary distribution, Markov chain Monte Carlo, areas and volumes, convergence of random walks on undirected graphs, random walks on undirected graph with unit edge weights, random walk in Euclidean space	13	0	0
4	**Machine learning**: Introduction, the perceptron algorithm, kernel functions, generalizing to new data, overfitting and uniform convergence, Occam's razor, regularization, online learning, support-vector machines, VC-dimension, boosting, stochastic gradient descent, deep learning	14	0	0
	Total	42	0	0

Learning Outcomes: Upon successful completion of this course, the student will:

1. have an understanding of basic mathematical concepts in data science, relating to linear algebra, probability, and calculus.
2. be able to employ methods related to these concepts in a variety of data science applications.
3. be able to adopt a rigorous and mathematical approach to solving problems in machine learning and data science.
4. be able to apply the mathematical concepts discussed over the duration of the course.

Teaching Methodology : 3 classroom lectures per week

Assessment Methods : Written examinations, quizzes and assignments

Text Books:

1. Avrim Blum, John Hopcroft and Ravindran Kannan, Foundations of Data Science, Cambridge University Press, February 29, 2020, ISBN-13: 978-1108485067

References: Will be prescribed by the instructor on a topic-by-topic basis.

Course Code and Title	CS5016: Computational Methods and Applications				
Programme	M.Tech in Computing and Mathematics	**Year of study**	1	**Semester**	2
Course credit	2-0-3-4				
Course category	PMT				
Prerequisite, if any	Familiarity with coding, calculus, probability and linear algebra				

Course Objectives: The main objective of this course is to introduce students to some of the key computational techniques used in modelling and simulation of real-world phenomena. This course is designed to expose students to techniques and methods from a variety of disciplines, not normally encompassed in a single course. Unlike other courses focused more specifically on algorithms, data structures or numerical analysis, this course emphasizes hands-on programming to understand a variety of scientific phenomena and computational methods.

Course Contents

A subset of the following topics will be covered in a typical offering of the course and the lecture/lab hours mentioned for each topic are only indicative of the actual hours.

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	Introduction to Python, plotting in Python, Monte Carlo simulations, sampling from a discrete distribution, estimating Pi using Monte Carlo simulations, verifying the prime number theorem, random text generation, introduction to Python's *scipy.stats* module	2	0	3
2	Concepts of object-oriented programming, basic algorithms for efficiently traversing and computing path length distributions of arbitrary networks, computing set of connected components using Breadth First Search, introduction to networks, small world networks, percolation, introduction to Pythons's *NetworkX* package	2	0	3
3	Introduction to linear system of equations, Jacobi and Gauss-Siedel iterative techniques for solving linear systems, interpolation and Lagrange polynomial, polynomial curve fitting, introduction to Python's *scipy.interpolate* and *numpy.linalg* (routines to solving equations and inverting matrices) modules	2	0	3
4	Numerical differentiation using forward-difference formula, elements of numerical integration, the trapezoidal rule, computing the Jacobian matrix, estimating position from accelerometer reading, introduction to Python's *scipy.integrate* sub-package	2	0	3
5	Discrete least squares approximations, orthogonal functions, orthogonal least square polynomials and least square approximation, discrete trigonometric approximations, fast Fourier transform (FFT). Introduction to Python's *numpy.polynomial* and *numpy.fft* packages	2	0	3

6	Introduction to ordinary differential equations (ODEs), theory of Initial-Value problems, Euler method for solving initial-value problems, simulating a simple pendulum, solving ODEs using Python's *scipy.integrate* module	2	0	3
7	Introduction to partial differential equations (PDEs), finite difference method for solving PDE, finding a root using the bisection method, fixed-point iteration, Newton's method, finding the fixed point of the dynamics of a single cardiac cell, simulation of the cardiac tissue PDE, using Python's visualization tools, introduction to Python's PDE solver	2	0	3
8	Introduction to matrix factorization, Cholesky decomposition, sampling from a continuous distribution, ratio-of-uniforms method, generating Gaussian random vectors, introduction to Python's *numpy.linalg* (routines for matrix decomposition) and *numpy.random* modules	2	0	3
9	Introduction to Eigen values and Eigen vectors, the power method, deflation methods, applications in page rank and mixing time of Markov chains, introduction to singular value decomposition (SVD), relation to the least square problem, sensitivity of SVD, computing SVD, application in image compression	4	0	6
10	Monte Carlo integration, importance sampling, introduction to Markov chain, Metropolis-Hastings algorithm, the Ising model, simulated annealing, Gibbs sampler	4	0	6
11	Introduction to convex optimization, constrained optimization, dual function, dual problem, gradient method with fixed and diminishing step size, rate of convergence and bounds on deviation from optimal value, introduction to Python's *CVXOPT* package	2	0	3
12	Introduction to k-SAT and colorability, the Davis-Putnam algorithm, introduction to Python's *python-sat* package, NP-completeness, number partitioning problem, recursive and dynamic programming	2	0	3
	Total	28	0	42

Learning Outcomes: Upon successful completion of this course, the student will:

1. have working knowledge of some popular and well-known computational methods.
2. be able to Write codes that use computational methods to numerically solve problems in a variety of disciplines.
3. Know about open source packages that implement popular computational methods.
4. be able to apply the mathematical concepts discussed over the duration of the course.

Teaching Methodology : 2 classroom lectures and one 3 hour lab session per week

Assessment Methods : Written exam, continuous lab assessment, assignments

Text Books: None

References:

1. Richard L. Burden and J. Douglas Faires, “Numerical Analysis,” Cengage Learning; 9th edition, January 1, 2015, ISBN-13: 978-8131516546
2. William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery, “Numerical Recipes: The Art of Scientific Computing,” Cambridge University Press; 3rd edition, September 6, 2007, ISBN-13: 978-0521880688
3. Chris Myers, “Computational Methods for Complex Systems: Notes and Exercises” [Available online at http://pages.physics.cornell.edu/~myers/teaching/ComputationalMethods]
4. Official documentation for Python 3 [Available online at https://docs.python.org/3]
5. Other reference materials will be prescribed by the instructor on a topic-by-topic basis.

Course Code and Title	CS6002: Combinatorial Optimization				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	Basic knowledge of Linear Algebra and Algorithms				

Course Objective
The objective of the course is to help students gain familiarity with fundamental combinatorial optimization techniques.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1.	**Introduction:** Network Flows, Ford-Fulkerson Method, Edmonds-Karp Algorithm, Linear Programming (LP), LP Formulation for Flows, LP Duality, Weak Duality Theorem, Complementary Slackness, Farkas' Lemma	10	0	0
2.	**Linear Programming based Optimization:** Minimum Spanning Trees, Shortest Paths, Totally Unimodular Matrices from Bipartite Graphs, Maximum Matching and Minimum Vertex Cover on Bipartite Graphs	10	0	0
3.	**Polyhedral Techniques:** Polyhedra and Polytopes of Linear Programs, The Matching and Perfect Matching Polytopes for General Graphs, The Independent Set Polytope, Polyhedral Characterization of Bipartite Graphs and Perfect Graphs	12	0	0
4.	**Selected Advanced Topics** like Semidefinite Programming based algorithms or Matroid based Algorithms	10	0	0
	Total	42	0	0

Learning Outcomes:
1. To familiarize with polyhedral combinatorics and linear programming techniques for designing efficient algorithms.
2. To be able to design and analyze algorithms using the techniques taught in the course.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Combinatorial Optimization: Polyhedra and Efficiency (Volumes A, B and C) by Alexander Schrijver. Springer-Verlag Berlin Heidelberg, 2003. ISBN: 978-3-540-44389-6.

References:

1. Combinatorial Optimization by William J. Cook, William H. Cunningham, William R. Pulleyblank, Alexander Schrijver. John Wiley & Sons, 2011. ISBN: 9781118031391.

2. Matching Theory by László Lovász, M. D. Plummer. American Mathematical Soc., 2009. ISBN 0821847597, 9780821847596.
3. Geometric Algorithms and Combinatorial Optimization by Martin Grötschel, László Lovász and Alexander Schrijver. Springer-Verlag Berlin Heidelberg, 1993. ISBN: 978-3-642-78242-8.

Course Code and Title	CS6003: Probabilistic Method				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	Basic knowledge of Discrete Mathematics and Probability				

Course Objective: The objective is to introduce students to the probabilistic method, a fundamental and powerful technique in theoretical computer science.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1.	**The basic probabilistic method**. Illustration of the basic method using Ramsey numbers, Dominating sets in graphs and Hypergraph coloring	8	0	0
2.	**First Moment Method.** Linearity of Expectation**.** Direct Examples De randomization using conditional expectations. Alterations. Moments,	10	0	0
3	**Second Moment Method.** Chebyshev inequalities, introduction to concentration. Higher moments and Chernoff bound. Dimensionality reduction.	12	0	0
4	Selected topics from Lovasz Local Lemma, Correlation Inequalities, Martingales and Tight Concentration	12	0	0
	Total	42	0	0

Learning Outcomes:

1. To familiarize with various random constructions that show the existence of the required combinatorial object.
2. To be able to apply the various techniques taught in the course to come up with existential and constructive proofs for the existence of various mathematical objects.
3. To familiarize with basic pseudo-random objects and derandomization techniques.
4. To be able to construct pseudo-random objects according to the needs of the problem and come up with efficient derandomization techniques for these objects.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. The Probabilistic Method by Noga Alon and Joel H. Spencer. Fourth Edition, Wiley, 2016. ISBN: 978-1-119-06195-3.

References:

1. Randomized Algorithms by Rajeev Motwani and Prabhakar Raghavan. Cambridge University Press, 2013. ISBN: 9780511814075.
2. Graph Colouring and the Probabilistic Method by Michael Molloy and Bruce Reed. Springer-Verlag Berlin Heidelberg, 2002. ISBN: 978-3-540-42139-9.

Course Code and Title	CS5614: Game Theory and Mechanism Design				
Programme	M.Tech in Computing and Mathematics	**Year of study**	**1 or 2**	**Semester**	**2 or higher**
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	Basic knowledge of calculus and probability				

Course Objectives: The main objective of this course is to introduce the fundamental tools of game theory and a few equilibrium concepts. During this course, we will also look at applications from a variety of disciplines and delve into some of the fascinating mathematics that underlies game theory. This course begins with an introduction into Game Theory. It then introduces strategic and extensive form games, and games with societal aspects to them. Subsequently, we will move onto understanding a few topics/concepts/ideas in mechanism design.

Course Content

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Introduction:** Motivation, theory of rational choice, utility functions	3	0	0
2	**Strategic form games:** Definition, examples, dominant strategy equilibria, pure strategy Nash equilibrium, mixed strategy Nash equilibrium, existence of Nash equilibrium, computation of Nash equilibria, potential games, two player zero sum games, minimax theorem	15	0	0
3	**Extensive form games:** Definition, examples, games of imperfect information, games of incomplete information, repeated games, the folk theorem of average payoffs, Nash equilibrium in repeated games with average payoff	7	0	0
4	**Cooperative games:** Transferable utility games, the core, the Shapley value, Nash bargaining	5	0	0
5	**Mechanism design:** Introduction, social choice functions and mechanisms, incentive compatibility and revelation theorem, the Gibbard-Satterthwaite impossibility theorem, quasilinear mechanisms, Vickrey-Clarke-Groves (VCG) mechanisms, auction, optimal mechanisms and Myerson auction	12	0	0
	Total	42	0	0

Learning Outcomes: Upon successful completion of this course, the student will:

1. be able to differentiate between different types of games Identify various equilibria within games
2. gain knowledge about fundamental concepts of non-cooperative and cooperative game theory
3. be able to explain precisely, and apply solution concepts to examples of games
4. be able to apply the mathematical concepts discussed over the duration of the course.

Teaching Methodology : 3 classroom lectures per week.

Assessment Methods : Written examinations, quizzes and assignments

Text Books:

1. Y. Narahari, "Game Theory and Mechanism Design: 4 (IISc Lecture Notes Series)," World Scientific Publishing Co Pvt Ltd, May 07, 2014, ISBN-13: 978-9814525046
2. Anna R. Karlin and Yuval Peres, "Game Theory, Alive," American Mathematical Society, Apr 27, 2017, ISBN-13: 978-1470419820 [Available Online].

References:

1. Siddharth Barman and Y. Narahari, "Game Theory Lecture Notes" [Available online at http://lcm.csa.iisc.ernet.in/gametheory]
2. Roger B. Myerson, "Game Theory: Analysis of Conflict," Harvard University Press, September 1997, ISBN-13: 978-0674341159.
3. Martin J. Osborne, "An Introduction to Game Theory," Oxford University Press, 2003, ISBN- 13: 978-0195128956.
4. D. Fudenberg and J. Tirole, "Game Theory," Indian Edition by Ane Books, 2005, ISBN-13: 978-8180520822.

Course Code and Title	CS5615 Coding Theory				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	Linear Algebra				

Course Objectives: This is a first-level graduate course on coding theory, which will introduce students to some of the classical methods in coding theory. While mathematical background on linear algebra and probability is assumed, coverage of necessary background on finite fields is included as part of the course. Through concrete examples of code construction, where simple, yet powerful mathematical tools are put to use, the course is expected to improve students' insights into the mathematical foundations.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	Binary block codes, Minimum distance, Error-detecting capability and error-correcting capability.	2	0	0
2	**Linear block codes:** Linear block codes, Generator matrix, Parity-check matrix. Dual code, Alternate characterizations of minimum distance for linear block codes, Repetition code, Single-parity-check code, Hamming Code, Bounds on Codes -- Singleton Bound, Hamming bound, Gilbert-Varshamov bound, Plotkin bound. Asymptotic version of these bounds.	8	0	0
3	**Decoding of linear block codes:** Maximum a-posteriori probability (MAP) decoding, Maximum likelihood (ML) decoding, Standard Array Decoding.	6	0	0
4	**Cyclic codes:** Review of Finite fields, Polynomial description of cyclic codes, generator and check polynomials, Roots of cyclic codes, BCH codes, Reed-solomon codes. Berlekamp-Welch decoding algorithm.	16	0	0
5	LDPC codes - binary expander codes, Sipser-Spielman decoding algorithm, introduction to iterative decoding.	10	0	0
	Total	42	0	0

Learning Outcomes: Upon successful completion of this course, students are expected to:

1. Use algebraic techniques to construct efficient codes
2. Identify the parameters of a given code the quality of a given code.
3. State and prove the limits on achievable code performance

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Ron M. Roth, *Introduction to Coding Theory*, Cambridge University Press, 2006, ISBN-13: 978-0521845045.
2. Tom Richardson, Rudiger Urbanke, *Modern Coding Theory*, Cambridge University Press, 2009, ISBN-13: 978-0521165761.

References:

1. J. H. van Lint, I*ntroduction to Coding Theory*, Springer, 1999, ISBN-13: 978-3540641339.
2. Shu Lin, Daniel J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, 1982, ISBN-13: 978-0132837965.
3. W. Cary Huffman, Vera Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, 2010, ISBN-13: 978-0521131704.
4. Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin, *On the locality of codeword symbols*, IEEE Transactions on Information Theory, 58(11) pp 6925–6934, 2012.
5. P. Vijaykumar, Error Correcting Codes, NPTEL Course, https://nptel.ac.in/courses/117/108/117108044/

Course Code and Title	CS5616 Computational Complexity				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	Basic understanding of algorithms				

Course Objective: This is a first course in computational complexity theory designed for graduate and motivated undergraduate students that introduces the basic time and space complexity classes, randomized complexity class and parallel complexity classes. The students also learn about typical problems in these classes. Through the classical results covered, students would learn about the relations between some of these complexity classes.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	Review of Basics Concepts. Turing Machine Model of Computation, Universal Turing Machines, Deterministic Turing Machine and Class P. Non-deterministic Turing Machines and Class NP. Satisfiability and Circuit Value Problems. Polynomial time verifiability, Polynomial-time reductions and NP-completeness. Cook-Levin Theorem and its proof, Other Time Complexity Classes co-NP, EXP and NEXP.	8	0	0
2	Diagonalization: Deterministic Time Hierarchy Theorem, Non-deterministic Time Hierarchy Theorem (without proof), Ladner's Theorem, Oracle Machines and Limits of Diagonalization - Baker-Gill-Solovay theorem.	6	0	0
3	Space Complexity: Space Hierarchy Theorem. Classes PSPACE, L, NL and co-NL, Savitch's Theorem, Configuration Graphs, Reachability Problems, Log space reductions, Completeness Results (PSPACE completeness and NL-completeness), NL=co-NL.	6	0	0
4	The Polynomial Hierarchy and Circuits: The Polynomial Hierarchy, Alternating Turing Machines, Circuit Model of Computation, Non-uniformity, Class P/poly, Karp-Lipton Theorem, Parallel Computations and Class NC, P-completeness.	8	0	0
5	Randomized Computation: Probabilistic Turing Machines, Classes RP, co-RP, BPP, ZPP. Sipser-Gacs theorem, Adleman's theorem.	6	0	0
6	Selected advanced topics (either one of the topics from the list below, or any other relevant topic of choice) : (a) Interactive proofs and the class IP, the class AM, IP=PSPACE, Introduction to PCPs and PCP theorem. Or (b) Complexity of counting – the class #P, #P- completeness, Valiant's theorem.	8	0	0
	Total	42	0	0

Learning Outcomes: Upon successful completion of this course, students are expected to

1. Know and state the relationship between various complexity classes studied in the course
2. State and explain the classical results in computational complexity theory

Teaching Methodology : Classroom lectures

Assessment Methods : Primarily through written examinations. Some weightage for student presentations is recommended, if the class size is small.

Text Books:

1. Sanjeev Arora and Boaz Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, 1 edition, 2009, ISBN-13: 978-0521424264.
2. Oded Goldreich, *Computational Complexity: A Conceptual Perspective*, Cambridge University Press, 1 edition, 2008, ISBN-13: 978-0521884730.

References:

1. Michiel Siper, *Introduction to the theory of computation*, Cengage, 3 edition, 2014, ISBN-13: 978-8131525296.
2. Luca Trevisan, *Lecture Notes on Computational Complexity*, https://people.eecs.berkeley.edu/~luca/notes/complexitynotes02.pdf
3. Christos H. Papadimitrou, *Computational Complexity*, Pearson, 1993, ISBN-13: 978-0201530827.

Course Code and Title	CS6004: Logic				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	-				

Course Objective

Logic is the systematic study of reasoning. Much like its role in providing a basis for reasoning in other areas of mathematics and philosophy, logic is the basis for reasoning about computational systems like programming languages, databases, etc. In turn, these applications in Computer Science has lead to formulation of deeper and wider forms of logic. This course aims at exposing the student into the broad area of Logic and its applications to Computer Science.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1.	**Preliminaries:** Why study formal logic? Reasoning about Logic and its circularity. The idea of meta language and models. Validity, Judgements and rules of inference. Soundness and Completeness.	10	0	0
2.	**Classical logic:** Propositional logic, predicate logic, notions of models, soundness, completeness. First order logic, Proof systems for first order logic and its soundness. Completeness of a given proof system for first order logic. Compactness for first order logic.	22	0	0
3.	**Selected Advanced Topics:** A subset of the following topics may be covered based on the instructor's choice. 1. Reasoning about imperative programs, Hoare logic. 2. Type theory and constructive logic. 3. Linear logic. 4. Model Logic	10	0	0
	Total	42	0	0

Learning Outcomes:

1. State and prove soundness and completeness of first order logic
2. Carry out formal proofs in various logics discussed in the course

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Text Books:

1. Michael Huth and Mark Ryan, Logic in Computer Science: Modelling and Reasoning about Systems, Cambridge University Press (2nd edition).ISBN-13: 978-0521543101.

References:

1. Arindama Singh, Logics for Computer Science, Second edition, PHI Learning, 2018. ISBN-13: 978-9387472433.
2. Raymond M. Smullyan, What is the Name of This Book?: The Riddle of Dracula and Other Logical Puzzles, Dover Publications. ISBN-13: 978-0486481982.
3. Irving M. Copi, Introduction to Logic, Pearson Education (14th edition). ISBN-13: 978-9332539617.
4. Jean-YvesGirard, Linear Logic: its syntax and semantics, Cambridge University Press, 1995.
5. Simon Thompson, Type Theory and Functional Programming, Addison-Wesley, 1991. ISBN-13: 978-0201416671.

Course Code and Title	CS5004: Approximation Algorithms				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	A first course in the design and analysis of algorithms				

Course Objective

This is an advanced elective which assumes familiarity with the basic techniques for algorithm analysis and design. This first two parts of this course will familiarise students to methods of dealing with hard problems using various approximation algorithm design techniques ranging from combinatorial techniques to more sophisticated LP based and probabilistic methods. It will also introduce students to different types of approximation algorithms like constant factor approximations, PTAS and FPTAS. The last part of the course gives an exposure to widely used reduction techniques for proving inapproximability of problems. The concepts and techniques will be introduced using a variety of sample illustrative problems.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Part 1 : Combinatorial Techniques:** Basics - Review of NP Hardness and reductions, approximation factor, technique of lower bounding optimum - simple approximation for vertex cover, greedy algorithm for set cover. Minimum spanning tree heuristic - metric Steiner tree and metric TSP. Layering technique - feedback vertex set. Strong NP-Hardness and pseudo-polynomial-time algorithms - knapsack. Dynamic programming - PTAS for Euclidean TSP, FPTAS for the knapsack, asymptotic PTAS for bin packing. Local search - max-cut, facility location problem.	15	0	0
2	**Part 2 : Algorithms using LP based and probabilistic techniques:** Simple rounding - max-SAT with small clauses, set cover, bin-packing. Random sampling and derandomization - max SAT, max-cut. Randomized rounding - set cover, multiway cut. Dual fitting - set cover. Primal-dual method - set Cover, steiner forest.	15	0	0
3	**Part 3 : Hardness of Approximation:** Techniques in proving the hardness of approximation: reductions from NP-complete problems, hardness of approximation of TSP, Reductions that preserve approximation, Reductions from probabilistically checkable proofs, Reductions based on unique games conjecture - tight hardness of max-cut and vertex cover.	10	0	0
	Total	40	0	0

Learning Outcomes:

1. To be able to identify the design technique used in a given approximation algorithm.
2. To be able to design and analyze approximation algorithms using various combinatorial approximation techniques taught in the course.
3. To be able to apply more sophisticated techniques like randomization and LP based optimization to computationally hard, yet simple optimization problems.
4. To be able to apply reduction techniques of proving hardness of approximation of simple combinatorial problems.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Textbook

1. Approximation Algorithms by Vijay V. Vazirani. Springer-Verlag Berlin Heidelberg, 2003. ISBN: 978-3-642-08469-0.

Reference

1. The Design of Approximation Algorithms by David P. Williamson and David B. Shmoys. Cambridge University Press, 2011. ISBN: 978-0521195270.
2. Approximation Algorithms for NP-Hard Problems by Dorit S. Hochbaum (Ed). PWS Publishing Company, 1997. ISBN: 978-0534949686.

Course Code and Title	CS5003: Parameterized Algorithms				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	A first course in the design and analysis of algorithms				

Course Objective

This is an advanced level course on the design and analysis of algorithms for computational intractable problems. There are two major objectives of this course. The first is to provide a comprehensive introduction to the field of parameterized algorithms and complexity. This includes an illustration of the key techniques for the design and analysis of fixed-parameter tractable algorithms and a demonstration of the notion of fixed-parameter intractability. The second goal of this course is to present some of the recent tools and techniques (like the framework for proving lower bounds, meta theorems, non-trivial applications of combinatorial min-max results and use of matroids) and prepare the students for research in this area of algorithm design.

Course Contents

S. No.	Topic	Lecture (hours)	Tutorial (hours)	Lab (hours)
1	**Introduction:** Review of NP-hardness, approaches to NP-hardness, motivation to parameterized algorithms, notion of fixed-parameter tractability, formal definitions of the key concepts.	2	0	0
2	**Basic toolkit:** Kernelization, branching and bounded-depth search trees, iterative compression, greedy localization. [7 lectures]	7	0	0
3	**Kernelization Techniques:** Crown decomposition, sunflower lemma, expansion lemma, kernels based on linear programming.	7	0	0
4	**Randomized methods:** Randomness in parameterized algorithms, the color coding technique, the chromatic coding technique, basic pseudo-random objects and derandomization.	8	0	0
5	**Treewidth:** Parameterized algorithms based on dynamic programming over tree decomposition, computing/approximating treewidth.	8	0	0
6	**Lower bounds:** Notion of fixed-parameter intractability, W-hierarchy, notion of parameterized reductions and examples, kernelization lower bounds.	8	0	0
	Total	40	0	0

Textbook

1. Parameterized Algorithms by M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Springer International Publishing, 2015. ISBN: 978-3-319-21275-3.

Reference

1. Fundamentals of Parameterized Complexity by R. Downey and M. R. Fellows. Springer-Verlag London, 2013. ISBN: 978-1-4471-5559-1.
2. Parameterized Complexity Theory by J. Flum and M. Grohe. Springer-Verlag Berlin Heidelberg, 2006. ISBN: 978-3-540-29953-0.

Learning Outcomes:
After the successful completion of this course, the students will be able to study the parameterized complexity of a given problem. In particular, they would be equipped to classify parameterized problems as being fixed-parameter tractable or intractable. They would be able to design parameterized algorithms (or show hardness) using vanilla versions or extended/modified variants of the various techniques taught in the course. They would also be able to state open problems in the area and their current complexity statuses.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Textbook

1. Parameterized Algorithms by M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Springer International Publishing, 2015. ISBN: 978-3-319-21275-3.

Reference

1. Fundamentals of Parameterized Complexity by R. Downey and M. R. Fellows. Springer-Verlag London, 2013. ISBN: 978-1-4471-5559-1.
2. Parameterized Complexity Theory by J. Flum and M. Grohe. Springer-Verlag Berlin Heidelberg, 2006. ISBN: 978-3-540-29953-0.

Course Code and Title	CS5820: Probability and Computing				
Programme	M.Tech in Computing and Mathematics	Year of study	1 or 2	Semester	2 or higher
Course credit	3-1-0-4				
Course category	PME				
Prerequisite, if any	-				

Course Objective

To introduce the power of probability theory and randomization techniques in computer science, with particular emphasis on analyzing algorithms that employ randomization.

Course Contents

Introduction: Events, probability spaces, random variables, expectation, conditional expectation, tail bounds including Markov's inequality, Chebyshev's inequality, Chernoff bounds. Sample applications include Karger's min-cut algorithm, randomized quicksort, and permutation routing on the hypercube.

Common Probability Distributions: Bernoulli and binomial random variables, geometric distribution, coupon collector's problem, Poisson distribution, normal distribution, power law distributions.

Hashing with Applications: Balls into bins, chain hashing, Bloom filters, pairwise independence, Chebyshev's inequality for pairwise independent variables, universal hash functions, perfect hashing, the count-min sketch, the power of two choices, cuckoo hashing.

Probabilistic Method: The counting argument, the expectation argument, sample and modify, the second moment method, the conditional expectation inequality, the Lovasz local lemma.

Markov Chains and Random Walks: Basic definitions, stationary distribution, variation distance and mixing time and their relation to graph spectrum, random walks on undirected graphs, the Monte Carlo method, the Metropolis algorithm, coupling.

Martingales: Basic definitions, stopping time, Wald's equation, Azuma-Hoeffding inequality.

Learning Outcomes: Students will be able to model probabilistic events using random variables and analyze simple probabilistic algorithms using probabilistic tools acquired in this course.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Textbooks:

1. Randomized Algorithms by Motwani and Raghavan, Cambridge University Press, 1995, ISBN-13: 978-0521474658.
2. Concentration of Measure for the Analysis of Randomized Algorithms, by Dubhashi and Panconesi, Cambridge University Press, 2009, ISBN: 978-1107606609
3. The Probabilistic method, by Alon and Spencer, 3rd edition, Wiley, 2008, ISBN 978-0-470-17020-5.

Course Code and Title	CS6001: Topics in Graph Theory				
Programme	M.Tech in Computing and Mathematics	Year of study	2	Semester	3 or 4
Course credit	3-0-0-3				
Course category	PME				
Prerequisite, if any	-				

Course Objective

This is a post-graduate level course in graph theory. The objective of the course is to prepare the students for research in graph theory and related areas. Learning Outcomes: By the end of this course, students will have a rigorous foundation in the key areas of graph theory. They will be able to state, explain, prove, and apply many fundamental results in these areas. They will be able to analyse these proofs and apply these proof techniques to solve similar problems. They will also get confident to follow long and intricate graph theoretic arguments. They will also be familiar with some of the major open problems in the area.

Course Contents

Matching, Covering and Packing : Matching in bipartite graphs, Matching in general graphs, Tree packing and arboricity, Tree covering theorem, Path covers, Gallai-Milgram theorem, Dilworth's theorem.

Connectivity : The structure of 2-Connected graphs, Block graph of a graph, The structure of 3-connected graphs, Menger's theorem Linking pairs of vertices, k-linked graphs, relationship between connectivity and k-linkedness.

Planarity : Plane graphs, Euler's formula, Planar graphs and Kuratowski's theorem, Plane dual and Abstract dual.

Colouring : Colouring of planar maps, five colour theorem, Colouring vertices, Greedy vertex colouring, Brook's thorem, Colouring edges, Konig's theorem, Vizing's theorem, List colouring, 5-choosability of planar graphs, Perfect graphs, Strong perfect graph theorem (statement only), Weak Perfect graph Theorem.

Hamilton cycles : Sufficient conditions, Hamilton cycles and degree sequences

A subset of topics from the following : Turan's extermal graph theorem, Minors and Hadwiger's conjecture, Tree-decompositions, Tree-width and Pathwidth, Forbidden minors, Random graphs and properties of almost all graphs, Ramsey theory for graphs.

Teaching Methodology : Classroom lectures

Assessment Methods : Written examination

Textbooks:

1. Reinhard Diestel, Graph Theory, Fifth Edition, Springer-Verlag Berlin Heidelberg, 2017, ISBN:978-3-662-53621-6.

2. Adrian Bondy and U. S. R. Murty, Graph Theory, Springer, 2008 edition, ISBN:978-1849966900.

References:

1. Douglas B. West, Introduction to Graph Theory, Second Edition, Prentice Hall, 2001, ISBN: 978-8120321427.