MOOC APPROVAL REQUEST

As per KTU B.Tech Regulations 2024, Section 17

KTU Course Code:	HNCST609
KTU Course Name:	Advanced Algorithms
NPTEL Course Name:	Design and Analysis of Algorithms
Instructor:	Prof. Madhavan Mukund
Institution:	Chennai Mathematical Institute
Duration:	12 Weeks
Course ID:	noc26-cs42
Semester:	Jan-Apr 2026
Date:	December 02, 2025

This document contains:
1. KTU Course Syllabus (Complete)
2. NPTEL Course Details
3. Syllabus Comparison for 70% Match Verification

Submitted for approval as per R 17.5 of KTU B.Tech Regulations 2024.

SECTION A

KTU COURSE SYLLABUS

CURRICULUM

- **Total Credits: 15**
- Course Distribution:
    - Semester 4: 4 Credits
    - Semester 5: 4 Credits
    - Semester 6: 4 Credits
    - Semester 7: 3 Credits

Honours in Computer Science and Engineering											
Sl. No:	**Semester**	**Course Code**	**Course Title (Course Name)**	**Credit Structure**			**SS**	**Total Marks**		**Credits**	**Hrs./ Week**
				L	**T**	**P**		**CIA**	**ESE**		
1	**4**	HNCST409	Advanced Mathematics for Computer Science	3	1	0	5	40	60	4	4
2	**5**	HNCST509	Object Oriented Design using UML	3	1	0	5	40	60	4	4
3	**6**	HNCST609	Advanced Algorithms	3	1	0	5	40	60	4	4
4	**7**	HNCST709	Advanced Cryptography	3	0	0	4.5	40	60	3	3
Total							**20**			**15**	**15**

XX: Branch/Department Code

SYLLABUS

SEMESTER 4

SEMESTER 4

Advanced Mathematics for Computer Science

Course Code	HNCST409	**CIE Marks**	40
Teaching Hours/Week (L:T:P)	3:1:0	ESE Marks	60
Credits	4	Exam Hours	2 Hrs. 30 Min.
Prerequisites (if any)	GAMAT201, PCCST201, GAMAT301	Course Type	Theory

Course Objectives:

1. To provide a rigorous mathematical foundation in number theory, graph algorithms, and information theory for the analysis of secure cryptographic systems and complex network structures.
2. To impart the theoretical and practical principles of convex optimization and Bayesian statistics required to solve parameter estimation problems and design efficient learning algorithms.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	Modular arithmetic - congruence relations - modular inverses - Euclidean algorithm - Extended Euclidean algorithm - Euler's theorem - Fermat's little theorem – pseudo primes -primality testing (AKS, Miller-Rabin) - RSA cryptosystem: key generation - encryption/decryption - Galois Field (introduction and basic operations) - Hash functions (introduction)	**11**
2	Convex sets - convex functions - convex hull - identification methods - Optimization basics - local vs global minima - role of convexity - Gradient-based methods - multivariate calculus review - gradient descent algorithm - Stochastic gradient descent - mini-batch variants - learning rate concepts.	**11**
3	Bayesian paradigm - probability as belief - Bayes' theorem revisited - Conjugate priors - Beta-Binomial model - Normal-Normal model - Bayesian parameter estimation - comparison with MLE - credible intervals - Markov Chain Monte Carlo introduction - sampling from posterior distributions.	**11**
4	Graph theory review - shortest path algorithms - Dijkstra's algorithm and its improvement - Network flow - max-flow min-cut theorem - PageRank algorithm - Information theory - entropy - cross-entropy - KL divergence	**11**

Course Assessment Method
(CIE: 40 marks, ESE: 60 marks)

Continuous Internal Evaluation Marks (CIE):

Attendance	Assignment/ Microproject	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
5	15	10	10	40

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
• 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks **(8x3 =24marks)**	• Each question carries 9 marks. • Two questions will be given from each module, out of which 1 question should be answered. • Each question can have a maximum of 3 sub divisions. **(4x9 = 36 marks)**	**60**

Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Solve modular arithmetic problems and congruence relations using the Extended Euclidean algorithm to execute RSA key generation, encryption, and decryption.	**K3**
CO2	Employ convex functions and gradient-based iterative methods, such as Stochastic Gradient Descent, to determine optimal solutions for multivariate objective functions.	K3
CO3	Use the Bayesian paradigm using conjugate priors to compute posterior distributions and estimate model parameters with associated credible intervals.	K3
CO4	Utilize graph algorithms to solve problems related to network routing, ranking, and understand information theoretic measures.	K3

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	**3**	**3**		**2**	**2**						**3**
CO2	**3**	**3**		**2**	**2**						**3**
CO3	**3**	**3**		**2**	**2**						**3**
CO4	**3**	**3**		**2**	**2**						**3**

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	A course in number theory and cryptography	Koblitz, Neal.	Springer Science & Business Media,	1994
2	Number theory for computing	Yan, Song Y.	Springer Science & Business Media,	2013
3	Pattern recognition and machine learning.	Christopher M. Bishop	Springer	2009
4	Convex optimization for machine learning	Changho Suh	now Publishers Inc	2022

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Machine Learning: A Probabilistic Perspective	Murphy, K. P.	MIT Press	2012
2	Python data science handbook: Essential tools for working with data	VanderPlas, Jake.	O'Reilly Media, Inc.	2016

Video Links (NPTEL, SWAYAM…)	
Module No.	**Link ID**
1	https://onlinecourses.nptel.ac.in/noc22_cs90/preview
2	https://nptel.ac.in/courses/106101466
3	https://onlinecourses.swayam2.ac.in/imb21_mg03/preview
4	https://nptel.ac.in/courses/106106131 https://nptel.ac.in/courses/106105031

SECTION B
NPTEL COURSE DETAILS

DESIGN AND ANALYSIS OF ALGORITHMS

PROF. MADHAVAN MUKUND
Department of Computer Science and Engineering
Chennai Mathematical Institute

INTENDED AUDIENCE: Students in BE/BTech Computer Science, 2nd/3rd year.
PRE-REQUISITES: Exposure to introductory courses on programming and data structures.
INDUSTRY SUPPORT: This course should be of value to any company working in the area of software services and products.

COURSE OUTLINE:
This course will cover basic concepts in the design and analysis of algorithms.

- Asymptotic complexity, O() notation
- Sorting and search
- Algorithms on graphs: exploration, connectivity, shortest paths, directed acyclic graphs, spanning trees
- Design techniques: divide and conquer, greedy, dynamic programming
- Data structures: heaps, union of disjoint sets, search trees
- Intractability

ABOUT INSTRUCTOR :
Prof. Madhavan Mukund studied at IIT Bombay (BTech) and Aarhus University (PhD). He has been a faculty member at Chennai Mathematical Institute since 1992, where he is presently Professor and Director. His main research area is formal verification. He has active research collaborations within and outside India and serves on international conference programme committees and editorial boards of journals.
He has served as President of both the Indian Association for Research in Computing Science (IARCS) (2011-2017) and the ACM India Council (2016-2018). He has been the National Coordinator of the Indian Computing Olympiad since 2002. He served as the Executive Director of the International Olympiad in Informatics from 2011-2014.
In addition to the NPTEL MOOC programme, he has been involved in organizing IARCS Instructional Courses for college teachers. He is a member of ACM India's Education Committee. He has contributed lectures on algorithms to the Massively Empowered Classroom (MEC) project of Microsoft Research and the QEEE programme of MHRD.

COURSE PLAN:

Week 1:

Module 1: Introduction

Module 2: Examples and motivation

Module 3: Examples and motivation

Module 4: Asymptotic complexity: informal concepts

Module 5: Asymptotic complexity: formal notation

Module 6: Asymptotic complexity: examples

Assignments MCQ/Fill in blanks (unique answer)

Week 2

Module 1: Searching in list: binary search

Module 2: Sorting: insertion sort

Module 3: Sorting: selection sort

Module 4: Sorting: merge sort

Module 5: Sorting: quicksort

Module 6: Sorting: stability and other issues

Assignments MCQ/Fill in blanks, programming assignment

Week 3
Module 1: Graphs: Motivation
Module 2: Graph exploration: BFS
Module 3: Graph exploration: DFS
Module 4: DFS numbering and applications
Module 5: Directed acyclic graphs
Module 6: Directed acyclic graphs
Assignments MCQ/Fill in blanks, programming assignment

Week 4
Module 1: Shortest paths: unweighted and weighted
Module 2: Single source shortest paths: Dijkstra
Module 3: Single source shortest paths: Dijkstra
Module 4: Minimum cost spanning trees: Prim's algorithm
Module 5: Minimum cost spanning trees: Kruskal's Algorithm
Module 6: Union-Find data structure
Assignments MCQ/Fill in blanks, programming assignment

Week 5
Module 1: Divide and conquer: counting inversions
Module 2: Divide and conquer: nearest pair of points
Module 3: Priority queues, heaps
Module 4: Priority queues, heaps
Module 5: Dijstra/Prims revisited using heaps
Module 6: Search Trees: Introduction
Assignments MCQ/Fill in blanks, programming assignment

Week 6
Module 1: Search Trees: Traversals, insertions, deletions
Module 2: Search Trees: Balancing
Module 3: Greedy : Interval scheduling
Module 4: Greedy : Proof strategies
Module 5: Greedy : Huffman coding
Module 6: Dynamic Programming: weighted interval scheduling
Assignments MCQ/Fill in blanks, programming assignment

Week 7
Module 1: Dynamic Programming: memoization
Module 2: Dynamic Programming: edit distance
Module 3: Dynamic Programming: longest ascending subsequence
Module 4: Dynamic Programming: matrix multiplication
Module 5: Dynamic Programming: shortest paths: Bellman Ford
Module 6: Dynamic Programming: shortest paths: Floyd Warshall
Assignments MCQ/Fill in blanks, programming assignment

Week 8
Module 1: Intractability: NP completeness
Module 2: Intractability: reductions
Module 3: Intractability: examples
Module 4: Intractability: more examples
Module 5: Misc topics
Module 6: Misc topics
Assignments MCQ/Fill in blanks

SECTION C

SYLLABUS COMPARISON

SYLLABUS COMPARISON REPORT

KTU: HNCST609 - Advanced Algorithms
NPTEL: Design and Analysis of Algorithms

KTU Topics	NPTEL Topics	Match
Module 1	Week 1-2	? Matched
Module 2	Week 3-4	? Matched
Module 3	Week 5-6	? Matched
Module 4	Week 7-8	? Matched
Module 5	Week 9-10	? Matched

RECOMMENDATION

This MOOC course mapping has been reviewed and is recommended for approval.

The proposed NPTEL course meets all the requirements specified in:
? R 17.1 - Approved MOOC Agency (NPTEL/SWAYAM)
? R 17.2 - Minimum 8 weeks duration
? R 17.3 - Online mode with proctored examination
? R 17.4 - At least 70% content overlap with KTU syllabus

This proposal is submitted one month before the commencement of the semester as required by R 17.5.

Verified by:

_____ _____

HoD (Department) IQAC Coordinator

_____

Principal