Department of Computer Science
Ashoka University

Discrete Mathematics

CS1104-1 & CS1104-2

Semester: Spring 2024

Faculty: Partha Pratim Das, `partha.das@ashoka.edu.in`, 98300-30880 (WhatsApp)

---

1 Overview

Discrete mathematics is the study of mathematical structures that can be considered *discrete* (in a way analogous to discrete variables, having a bijection with the set of natural numbers) rather than *continuous* (analogously to continuous functions). A related but somewhat different interpretation of the phrase *discrete mathematics* denotes all those branches of mathematics that deal with discrete entities as opposed to continuous entities. For example, a typical undergraduate course on calculus will not be considered discrete in this sense, whereas probability theory is something that can come in both discrete and continuous flavors.

Sound understanding of *Discrete Mathematics* is critical for all areas of Computer Science. Hence this course attempts to lay the foundation for a curriculum in CS. Many of the thematic areas as covered like – combinatorics (counting methods), graph theory, number theory, information theory, etc. – are deep and vast subject areas on their own. Interested students are encouraged to follow up on any area/s they might find interesting. However, everyone interested to pursue CS must grasp the fundamentals in all these well.

2 Objectives / Learning Outcomes

To lay the mathematical foundations for computational thinking for the core and elective courses including ***Data Structures and Algorithm***, ***Design and Analysis of Algorithms***, ***Programming Languages and Translation***, ***Theory of Computation***, ***Design Practices in CS***, ***Introduction to Machine Learning***, ***Data Science and Management***, *and* ***Information Security*** *to build on.*

The course:

1. introduces students to mathematical structures, tools, techniques, and frameworks for algorithmic and computational thinking
2. illustrates ways for problem formulation, representation, and analysis using mathematical models
3. elucidates how mathematical rigour and formalism ensure soundness and completeness of computation

The emphasis throughout the course is on teaching how various concepts and results of mathematics are used in correct, efficient, and robust computations. *It trains the students to think and reason about computation in a programming language-agnostic manner. It complements the course on* ***Introduction to Computer Science*** *from mathematical perspective and thinking.*

3 Pre-requisites

The course expects familiarity with class 12 level mathematics (sets, relations, functions, basic logic and truth tables, basic counting, the principle of mathematical induction, and calculus).

4 Module Coverage

The course comprises 9 modules in 4 parts:

Part 1: Models and Proofs

1. *Propositional and Predicate Logic*
2. *Proof Techniques*

Part 2: Mathematical Structures

3. *Basic Structures: Sets, Functions, Relations, Sequences and Matrices*

Part 3: Computational Problem Solving

4. *Problem Solving with Recursive Decomposition*
5. *Graphs and Trees*
6. *Models of Computation*

Part 4: Techniques for Analyses

7. *Counting*
8. *Number Theory*
9. *Probability and Information Theory*

The detailed outline of topics in every module is presented in the next section.

5 Detailed Outline of Topics in Modules

This course would introduce the notion of a model of computation, and issues of correctness and efficiency in simple settings. It would emphasise that algorithm and program design are not ad-hoc and trial-and-error based methods. The course would also introduce the notion of states, invariants, logic, and almost everything from the algorithm design and logic baskets of various CS courses with simple examples. It would also touch up on the notions of abstract data types and data structures. This and the Data Structures and Algorithm *courses would use the same style of treatment – with emphasis on correctness using induction and invariants, and on analysis of efficiency – on more complex examples.*

The course comprises 9 modules in 4 parts:

Part 1: Models and Proofs

The first part of the course deals with the modeling of the real-world using mathematical logic, and then solving various problems using rules of inference. It also introduces representations of truth statements like theorems, lemmas, corollaries etc. and various techniques to prove or refute them. These lay the foundation of thinking about soundness and correctness of computations we build.

1. **Propositional and Predicate Logic**
   - *Propositional Logic*: Propositions, Operators, Applications, Equivalences, and Satisfiability.
   - *Predicate Logic*: Predicates, Quantification, Precedence, Bound & Free Variables, and Nested Quantifiers.
   - *Arguments in Logic*: Rules of Inference (for Propositions), Resolution, and Rules of Inference with Quantification
   - *Logic Programming in Prolog*: Modeling and Problem Solving with Prolog
2. **Proof Techniques**:
   - *What is a proof*: Types of truth statements, Fallacies
   - *Proof Methods and Strategy*: Direct, Indirect – Contraposition & Contradiction, Induction and Others, Soundness & Completeness

Part 2: Mathematical Structures

The second part of the course deals with basic mathematical structures necessary for understanding, modeling, and structuring computational processes. Starting with sets, notions of functions, relations, and matrices are developed for formal reasoning in computations. Handling of infinity is elucidated with infinite sets and sequences including sets of integers, rational numbers, reals, and Fibonacci sequence.

3. **Basic Structures: Sets, Functions, Relations, Sequences and Matrices**
    - *Sets*: Basic Notions, Operators & Representation of Sets, and Fuzzy & Rough Sets
    - *Functions*: Basic Notions & Types, and Composition & Graphs
    - *Finite and Infinite Sets*: Countability of Sets, and Theory of Infinity
    - *Relations*: Equivalence Class, and Partial Order (Poset)
    - *Matrices*: Basic notions & Operations of matrices
    - *Sequences and Summations*: Arithmetic & Geometric Progressions, and Recurrence Relations & Summations

Part 3: Computational Problem Solving

The third part of the course addresses three major aspects of computational problem solving – problem decomposition and refinement, problem representation, and computational models. Recursive decomposition with refinement by solution space exploration is introduced through a set of simple problems of graded complexity. These will lead to the algorithmic strategies in the *Design and Analysis of Algorithms* course later. Graphs and Trees are introduced as widely used models for problem representation in general and will lead to various data structure in the *Data Structures and Algorithm* course later. Finally, languages, finite automata, and Turing machine are introduced to understand the models of computation. These will be proliferated later in *Program Languages and Translation* and *Theory of Computation* courses.

4. **Problem Solving with Recursive Decomposition**
    - *Solution Space Exploration with Recursive Decomposition*: Demonstration with simple problems
    - *Analysis of Algorithms*: Time & Space Complexity, and Counting Models
    - *Growth of Functions*: Asymptotic Analysis, and Master Theorem
    - *Recursion and Iteration*: Tail recursion, Fibonacci by recursion, and Fibonacci by memoization / iteration
    - *Induction and Recursion*: Recursive Definitions, Structural Induction, Program Correctness
5. **Graphs and Trees**
    - *Graphs*: Graphs and Digraphs, Basic Notions, Representations, Graph Traversals, Special Graphs, Operations, and Properties
    - *Trees*: Tree Traversal, and Spanning Tree
    - *Graph Problems*: Problem solving with graphs
6. **Models of Computation**
    - *Language Models*: Basic Notions, and Chomsky Hierarchy of Languages
    - *Automata Models*: History of Automata Theory, Finite Automata (DFA), Language recognizers, and Turing machine

Part 4: Techniques for Analyses

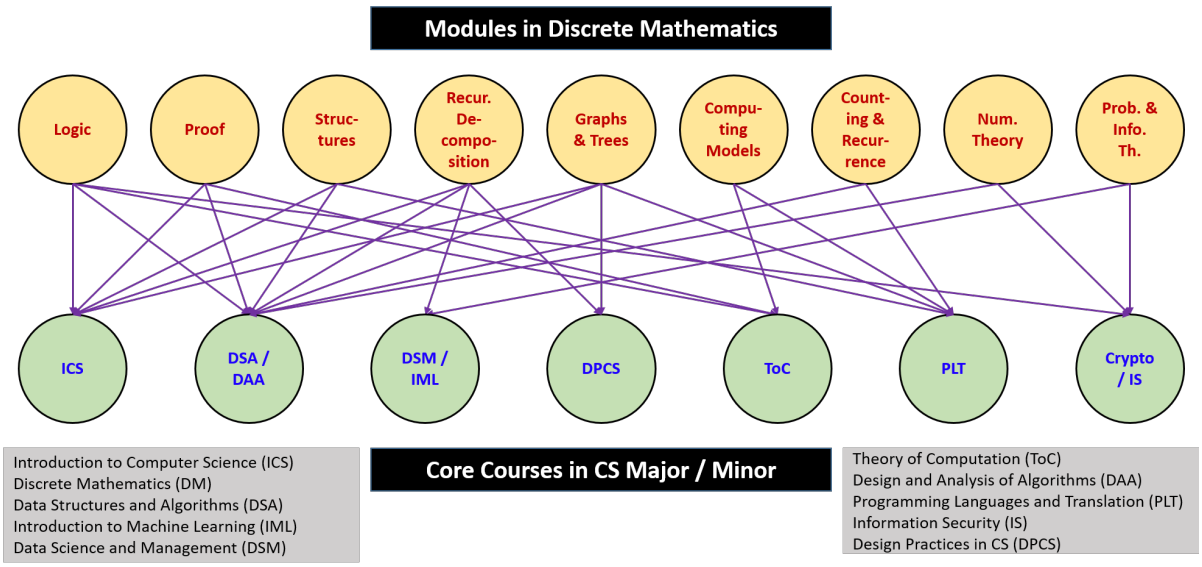
The fourth and final part of the course addresses a number of useful techniques and mathematical notions that help in analyses of computations. Counting, recurrence relations, modular arithmetic, discrete probability and information theory provide foundational mathematical tools in computing.

7. **Counting**
    - *Counting Principles*: Counting Examples, Counting Rules, Tree Diagrams, Pigeonhole Principle, Permutations, Combinations, Binomial & Multinomial Coefficients, and Practice Problems
    - *Recurrence Relations*: Linear Homogeneous Recurrence, Linear Non-Homogeneous Recurrence, Generating Functions of Sequences – Manipulations, Closed Forms, and Practice Problems
8. **Number Theory**
    - *Divisibility and Modular Arithmetic*: Division, Division Algorithm, Modular Arithmetic, and Arithmetic Modulo m
    - *Integer Representations and Algorithms*: Algorithms for Integer Operations, and Modular Exponentiation
    - *Primes and Greatest Common Divisors*: Primes, and Greatest Common Divisors & Least Common Multiples
    - *Congruences*: Solving Congruences, and Applications of Congruences
9. **Probability and Information Theory**
    - *Discrete Probability*: Finite Probability, Complements & Unions, Probabilistic Reasoning, Probability Theory, Bayes' Theorem, and Expected Value & Variance
    - *Information Theory*: What is Information Theory?, Information and Entropy, Joint & Conditional Entropy, Relative Entropy & Mutual Information, Chain Rules for Entropy, Relative Entropy, & Mutual Information, and Huffman Code

5.1 Module to Course Mapping

The diagram in Figure 1 shows the mapping between various modules of this course and core and elective courses for Major / Minor in CS. We map a module to a course if the learning of the module is used and / or developed further in the course.

Figure 1: Module to Course Mapping



6 Textbooks and Readings

- Slides with links to online resources
- *Discrete Mathematics and its Applications*, 7^{th} Ed. by Kenneth H. Rosen (*Primary Textbook*)
- *Concrete Mathematics – A Foundation for Computer Science*, 2^{nd} Ed. by Ronald L. Graham, Donald E. Knuth, and Oren Patashnik
- *Discrete Mathematics for Computer Scientists and Mathematicians* by Joe L. Mott, Abraham Kandel, and Theodore P. Baker.
- *Applied Combinatorics* by Fred Roberts and Barry Tesman.
- *Graph Theory with Applications to Engineering and Computer Science* by Narsingh Deo.
- *Introduction to Graph Theory* by Douglas B. West.
- *Elements of Information Theory*, 2^{nd} Ed. by Thomas M. Cover and Joy A. Thomas

No book discusses all the topics. *So it is important to attend all the lectures by the instructor. You are also expected to carefully study the solved problems that will periodically be posted.*

7 Lecture Distribution

Module # and Module Title	# Lectures
Part 0	
0. Course Information and Overview	1
Part 1	
1. Propositional and Predicate Logic	3
2. Proof Techniques	2
Part 2	
3. Basic Structures: Sets, Functions, Relations, Sequences and Matrices	4
Part 3	
4. Problem Solving with Recursive Decomposition	3
5. Graphs and Trees	3
6. Models of Computation	2
Part 4	
7. Counting	4
8. Number Theory	1
9. Probability and Information Theory	2
Total	**25**

7.1 Discussion Sessions

A number of discussion sessions would be conducted by the Teaching Fellow to supplement and complement the contents in the module lectures. Some of the identified DSs include:

- Logic Programming in Prolog
- Selected Graph Algorithms
- Recognisers for Regular Languages and Regular Expressions – NFA and DFA
- Generating Functions to solve Homogeneous and Non-Homogeneous Linear Recurrences

More will be decided on the need basis.

8 Learning Management System

We will be using *Google Classroom* as the learning management system for this course. You will automatically be added to it for this course via the *Ashoka LMS*. All the course material, including lecture presentations, and assignments will be made available through the Google Classroom. In addition, *we will be accepting student submissions only through Google Classroom links*.

9 Class Schedule

Classes will be held in the scheduled hours as follows.

CS1104-1

- *Every Monday, from 3:00pm to 4:30pm*, in AC02-110 (LT)
- *Every Wednesday, from 3:00pm to 4:30pm*, in AC02-110 (LT)

CS1104-2

- *Every Monday, from 4:40pm to 6:10pm*, in AC04-LR-005
- *Every Wednesday, from 4:40pm to 6:10pm*, in AC04-LR-005

Some classes may be held online while the rest will be physical. We will use *Zoom for online classes*. The link will be shared before the class.

In addition, we will have *a common slot for both sections on every week earmarked for conducting quizzes, mid-term examination, and any additional discussions and / or tests* we may need. This will be decided from the following list of slots in consultations with both sections once the semester starts:

- *Tuesday, from 8:30am to 10:00am*
- *Tuesday, from 8:00pm to 9:30pm*
- *Wednesday, from 8:30am to 10:00am*
- *Wednesday, from 8:00pm to 9:30pm*

The room to meet will be decided on the need basis.

10 Attendance

Attendance in every class is mandatory. Please note that the potential actions on getting low on attendance:

- If your attendance falls below 80% at any time during the semester or if you are absent from two consecutive classes, you will be reported to the Head, Computer Science.
- If your attendance falls below 60% at any time during the semester or if you are absent from four consecutive classes, you will be reported to the Head, Computer Science and Dean, Academic Affairs, and de-registration from the course would be sought.

The exemption to the above will be only on medical ground:

- Every missed attendance on medical ground must be corroborated by written certificate of incapacitation or prescription of rest issued from the infirmary at Ashoka. No other medical certificate will be accepted.

  The certificate justifying the dates of the classes missed must be mailed prior to next class.
- No missed attendance will be allowed for representing in sports or other events, for taking examinations elsewhere, or on any other ground.

11 Consultation and Academic Support

The course staff (instructor, TF and TAs) would be regularly available for consultation and academic support.

- **Offline Consultation**: The course staff would be available on email on a daily basis (almost on all days). Please contact using the email of teaching team as given in Section 12. *Emails to individual course staffs may be ignored*.
- **Face-to-face Consultation with the Instructor**: The instructor will hold an **Office Hours (OH)** regularly by appointment based on the mutual convenience of the instructor and the students. The TF / TAs will coordinate the schedule. The office hours may be physical or online.
- **Face-to-face Consultation with the TF / TAs**: Office hours with the TF / TAs will be held on a need basis and based on the mutual convenience of the TF / TAs and the students. Often TF / TAs would call for OH to discuss evaluations of assignments and / or quizzes.

12 Teaching Team

While the TF for the course has been finalized, the TAs are yet to be assigned. Hence, the following team tables will be updated closer to the start of the semester.

CS1104-1

Sl.#	Name	Email	TA / TF
1	Adwaiya Srivastav	adwaiya.srivastav_tf@ashoka.edu.in	TF
2	TBD		TA
3	TBD		TA
4	Partha Pratim Das	partha.das@ashoka.edu.in	

Please do not call or WhatsApp the team members
Send mail to cs1104-1-staff@ashoka.edu.in *when you need any help*

CS1104-2

Sl.#	Name	Email	TA / TF
1	Adwaiya Srivastav	adwaiya.srivastav_tf@ashoka.edu.in	TF
2	TBD		TA
3	TBD		TA
4	Partha Pratim Das	partha.das@ashoka.edu.in	

Please do not call or WhatsApp the team members
Send mail to cs1104-2-staff@ashoka.edu.in *when you need any help*

13 Evaluation and Grading

The following is a tentative guideline for evaluations and the grading policy. It may undergo change that will be announced in the class and / or Google Classroom.

1. **Assignments**: There will be an assignment on each module to solve mathematical problems and/or write mathematical proofs. All the assignments will
   - be combined to contribute **20%** to your final score.
   - not carry the same credits.
   - be released through the Google Classroom.
   - be take-home with online submission through the Google Classroom.
   - be individual. There will be no group assignment.
2. **Quizzes**: There will be three in-class quizzes that will require you to solve mathematical problems and/or write mathematical proofs. All the quizzes will
   - be combined to contribute **30%** to your final score.
   - carry the same credits.
   - be conducted physically in the classroom with physical submission.
3. **Mid-term Exam**: Mid-term Exam will
   - contribute **20%** to your final score.
   - be conducted physically in the classroom with physical submission.
4. **End-term Exam**: End-term Exam will
   - contribute **30%** to your final score.
   - be conducted physically in the classroom with physical submission.

13.1 Score to Grade Conversion

The grading in the course will be relative. Final grades will be assigned in line of the *Indicative Percentage Band to Grade Conversion* as shown in Table 1. Depending on the Bell Curve, the letter grade boundaries will be adjusted at the time of final grading.

Table 1: Numerical Score to Letter Grade Conversion Policy

Final Score	**Letter Grade**
$x \leq 50$	F
$50 < x \leq 53$	D–
$53 < x \leq 56$	D
$56 < x \leq 59$	D+
$59 < x \leq 62$	C–
$62 < x \leq 66$	C
$66 < x \leq 70$	C+

Final Score	**Letter Grade**
$70 < x \leq 75$	B–
$75 < x \leq 80$	B
$80 < x \leq 86$	B+
$86 < x \leq 93$	A–
$x > 93$	A

x denotes the aggregate numerical score accumulated by you over the entirety of the course duration according to the policies outlines above.

Students need to achieve a minimum grade of "B" in this courses (and in Introduction to Computer Science) to be able to pursue major or minor or concentration in CS.

14 Submission and Compensation

We are aware of the fact that you're juggling multiple activities and the assigned deadlines may not always be favorable. In order to meet deadlines, it would be best for you to start on your assignments as soon as they are handed out. Starting working on an assignment on the day of the submission is the worst idea you could possibly come up with. Historically, this has resulted in scores that are below average performance. Starting your work late also creates conditions for plagiarism, which is something that you should definitely avoid – at all costs. You are expected to submit the assignments before the assigned deadline. We will follow the following policies:

1. **Mode of Submission**: All assignments must be submitted electronically through the Google Classroom for the course. *No email based submissions will be considered.*
2. **Format of Submission**: All assignments must be clearly hand-written, scanned and submitted as a single PDF named as `A<assignment #>_<your first name #>_<your Ashoka ID #>.pdf`.
3. **Late Submission**: Late submissions are possible.
   - An assignment is counted as late, as soon as the clock for the assignment submission rolls over.
   - For every extra day that you take to submit, the assignment will lose 20% of its value.
   - Once the assignment loses 100% of its value, it will not be graded.
4. **No Extension for Submission Deadline**: *No extension for the submission deadline of assignments will be allowed for any assignment. The deadline once announced will be the final. So the teaching team will appreciate that you do not waster your and their time by asking for the same.*
5. **Extension for Submission Deadline on Individual Emergency**: Extensions may be granted individually only in case of genuine emergencies, for example, medical problems, family-related issues, etc. All such extensions will solely be at the instructor's discretion.
   - Every request for extension on a medical ground must be corroborated by written certificate of incapacitation or prescription of rest issued from the infirmary at Ashoka. No other medical certificate will be accepted.
   - No extension will be allowed for representing in sports or other events.

6. **Compensation for Quiz / Mid-term Exam**: Only one compensation quiz may be allowed in the entire duration of the course to compensate for one or more quiz and / or mid-term exam. That is, if you miss two or more quizzes and / or mid-term exam, you will be considered for one compensation quiz only. All such extensions will solely be at the instructor's discretion.
    - The compensation may be granted only in case of medical emergency.
    - Every request for compensation on a medical ground must be corroborated by written certificate of incapacitation or prescription of rest issued from the infirmary at Ashoka. No other medical certificate will be accepted.
    - No compensation will be allowed for representing in sports or other events.
7. **Compensation for End-term Exam**: No compensation for end-term exam may be provided unless expressly recommended by the Head of the Department and the Dean, Academic Affairs.

15 Cheating and Plagiarism

- For any assignment, feel free to refer to the lectures or the primary reading material. Looking up or seeking help from every other source – including the internet and ChatGPT[1], is allowed but copying answers will be considered instances of plagiarism.
- For the assignments, you are be allowed to discuss and collaborate on the problems with your peers in the class[2]. However, any work that you submit should be written up independently and individually. One rule of thumb that you may follow is:

    > *While discussing something with your peers, do not use any pen and paper (or laptop). That is, do not write down anything during the discussions (apart from on a blackboard). Allow yourself to take only mental notes, and at a later point in time, when you are in the solitude of your own room, work out the mental notes on paper.*

- We will have zero tolerance for cheating – be it an assignment or an exam (quiz / mid-term / end-term). Any evidence violation of academic integrity or of plagiarism will result in a 0 on that graded component and a plagiarism flag. Students with two plagiarism flags will be given an F grade for the entire course. No exceptions will ever be made. In addition, you will be reported to the OAA for academic integrity violation.
- If you have any, however small, questions about what counts as plagiarism – and what does not – please contact the instructor. *But I didn't know* `< insert-the-thing-you-did >` *was also counted as plagiarism* will not be considered an acceptable excuse.

16 Calendar

The tentative calendar of evaluation events is given below. This may change in consultations with you.

Week	Modules	Event		
		Release	Submission	Conduct
W01	M00, M01	A1		
W02	M01			
W03	M02	A2	A1	
W04	M03	A3		Q1 [M01-02]
W05	M03		A2	
W06	M04	A4	A3	
W07	M04, M05	A5		Q2 [M03]
W08	Break W			
W09	M05		A4	MS [M01-04]
W10	M06	A6	A5	
W11	M07	A7	A6	
W12	M07			Q3 [M05-06]
W13	M08, M09	A8	A7	
W14	M09	A9	A8	
W15	Reading W		A9	Q4 [M01-09]
W16	Exam W			ES [M01-09]

[1] TAs will check similarity with ChatGPT-generated solutions for every assignment / exam. So, ...
[2] Please mention the people you worked with at the top left of your assignment under a **Collaborators** heading.