

**Ministry of Education (MINEDU)**  
Syllabus 2021-I

**1. COURSE**

CS1D2. Discrete Structures II (Mandatory)

**2. GENERAL INFORMATION**

<b>2.1 Credits</b>	:	4
<b>2.2 Theory Hours</b>	:	2 (Weekly)
<b>2.3 Practice Hours</b>	:	2 (Weekly)
<b>2.4 Duration of the period</b>	:	16 weeks
<b>2.5 Type of course</b>	:	Mandatory
<b>2.6 Modality</b>	:	Face to face
<b>2.7 Prerequisites</b>	:	CS1D1. Discrete Structures I. (1 <sup>st</sup> Sem)

**3. PROFESSORS**

Meetings after coordination with the professor

**4. INTRODUCTION TO THE COURSE**

In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

**5. GOALS**

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student applies efficient travel strategies to be able to search data in an optimal way.
- That the student uses the various counting techniques to solve computational problems.

**6. COMPETENCES**

- a) An ability to apply knowledge of mathematics, science. ( **Usage**)
- j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. ( **Usage**)
- a) An ability to apply knowledge of mathematics, science. ( **Familiarity**)
- b) An ability to design and conduct experiments, as well as to analyze and interpret data. ( **Usage**)
- i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. ( **Familiarity**)
- a3) (3)
- a9) (9)
- a13) (13)
- a14) (14)
- j1) (1)
- j2) (2)

**7. TOPICS**

**Unit 1: Digital Logic and Data Representation (10)****Competences Expected: a,b,i****Topics**

- Reticles: Types and properties.
- Boolean algebras.
- Boolean Functions and Expressions.
- Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form.
- Logical gates.
- Circuit Minimization.

**Learning Outcomes**

- Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment].
- Explain the algebraic structures of reticulum and its types [Assessment].
- Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment].
- Explain the properties that satisfies a Boolean algebra [Assessment].
- Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment].
- Find the canonical forms of a Boolean function [Assessment].
- Represent a Boolean function as a Boolean circuit using logic gates [Assessment].
- Minimize a Boolean function. [Assessment].

**Readings :** [Ros07], [Gri03]

<b>Unit 2: Basics of Counting (40)</b>	
<b>Competences Expected: a</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Counting arguments               <ul style="list-style-type: none"> <li>– Set cardinality and counting</li> <li>– Sum and product rule</li> <li>– Inclusion-exclusion principle</li> <li>– Arithmetic and geometric progressions</li> </ul> </li> <li>• The pigeonhole principle</li> <li>• Permutations and combinations               <ul style="list-style-type: none"> <li>– Basic definitions</li> <li>– Pascal’s identity</li> <li>– The binomial theorem</li> </ul> </li> <li>• Solving recurrence relations               <ul style="list-style-type: none"> <li>– An example of a simple recurrence relation, such as Fibonacci numbers</li> <li>– Other examples, showing a variety of solutions</li> </ul> </li> <li>• Basic modular arithmetic</li> </ul>	<ul style="list-style-type: none"> <li>• Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity]</li> <li>• Apply the pigeonhole principle in the context of a formal proof [Familiarity]</li> <li>• Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity]</li> <li>• Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity]</li> <li>• Solve a variety of basic recurrence relations [Familiarity]</li> <li>• Analyze a problem to determine underlying recurrence relations [Familiarity]</li> <li>• Perform computations involving modular arithmetic [Familiarity]</li> </ul>
<b>Readings :</b> [Gri97]	

<b>Unit 3: Graphs and Trees (40)</b>	
<b>Competences Expected: a</b>	
<b>Topics</b>	<b>Learning Outcomes</b>
<ul style="list-style-type: none"> <li>• Trees               <ul style="list-style-type: none"> <li>– Properties</li> <li>– Traversal strategies</li> </ul> </li> <li>• Undirected graphs</li> <li>• Directed graphs</li> <li>• Weighted graphs</li> <li>• Spanning trees/forests</li> <li>• Graph isomorphism</li> </ul>	<ul style="list-style-type: none"> <li>• Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity]</li> <li>• Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity]</li> <li>• Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity]</li> <li>• Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity]</li> <li>• Explain how to construct a spanning tree of a graph [Familiarity]</li> <li>• Determine if two graphs are isomorphic [Familiarity]</li> </ul>
<b>Readings :</b> [Joh99]	

## 8. WORKPLAN

### 8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

### 8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

### 8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

## 9. EVALUATION SYSTEM

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BASIC BIBLIOGRAPHY

- [Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.
- [Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.
- [Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.
- [Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.