TMM023- DISCRETE MATHEMATICS

(March 2021)

Typical Range: 3-4 Semester Hours

A course in Discrete Mathematics specializes in the application of mathematics for students interested in information technology, computer science, and related fields. This college-level mathematics course introduces students to the logic and mathematical structures required in these fields of interest.

TMM023 Discrete Mathematics introduces mathematical reasoning and several topics from discrete mathematics that underlie, inform, or elucidate the development, study, and practice of related fields. Topics include logic, proof techniques, set theory, functions and relations, counting and probability, elementary number theory, graphs and tree theory, base-*n* arithmetic, and Boolean algebra.

To qualify for TMM023 (Discrete Mathematics), a course must achieve all the following essential learning outcomes listed in this document (marked with an asterisk). In order to provide flexibility, institutions should also include the non-essential outcomes that are most appropriate for their course. It is up to individual institutions to determine if further adaptation of additional course learning outcomes beyond the ones in this document are necessary to support their students' needs. In addition, individual institutions will determine their own level of student engagement and manner of implementation. These guidelines simply seek to foster thinking in this direction.

1. **Formal Logic** – Successful discrete mathematics students are able to construct and analyze arguments with logical precision. These students are able to apply rules from logical reasoning both symbolically and in the context of everyday language and are able to translate between them.

   The successful Discrete Mathematics student can:

   1.1. Propositional Logic:

   1.1a. Translate English sentences into propositional logic notation and vice-versa. *

   Sample Tasks:

   - Determine if a sentence is a proposition.
   - Construct the negation of a statement, the conjunction of two statements, and the disjunction of two statements.
   - Make use of the symbols $\neg$, $\rightarrow$, $\vee$, $\wedge$, etc.
   - Translate English sentences into propositional logic using appropriate notation, such as "John is healthy and wealthy but not wise".

   1.1b. Construct truth tables for statements involving the following logical connectives: negation, conjunction, disjunction, conditional, and biconditional.*

   Sample Tasks:

   - Construct the truth tables for a given compound proposition.
   - Determine if two compound statements are logically equivalent using truth tables.

1.1c. Apply De Morgan's Laws to find negations of statements. *

Sample Tasks:

- Negate an *and* statement.
- Negate an *or* statement.
- Use De Morgan's Laws to negate statements involving intervals such as $2 < x < 7$.

1.1d. Define and use these terms: conditional statement, converse, inverse, contrapositive, biconditional, necessary and sufficient conditions, tautology, contingency, and contradiction.*

Sample Tasks:

- Find the negation, converse, inverse and contrapositive of a given conditional statement.
- Interpret and translate English sentences that express necessary and sufficient conditions into formal logic, such as , "Catching the 8:05 bus is a sufficient condition for me being on time for work." or "A necessary condition for this computer program to be correct is that it not produce error messages during translation."
- Determine if a given compound proposition is a tautology, contradiction, or contingency, such as $p \vee \neg p, p \wedge \neg p$.

1.1e. Apply standard logical equivalences to simplify propositions, and be able to prove that two logical expressions are or are not logically equivalent. *

Sample Tasks:

- Use basic laws of equivalence, such as Commutative laws, Associative laws, Distributive laws, De Morgan's laws, Identity laws, Double Negation law, Idempotent laws, Universal bound laws, and Absorption laws, to simplify propositions.

1.1f. Determine if a logical argument is valid or invalid. Apply standard rules of inference including Modus Ponens, Modus Tollens, Generalization, Specialization, Conjunction, Transitivity, and Elimination. Recognize fallacies such as the Converse Error and the Inverse Error. *

Sample Tasks:

- Determine whether a given argument form is valid or invalid by constructing a truth table.
- Determine whether a given argument form is valid or invalid by applying standard rules of inference.
- Identify errors in reasoning that result in an invalid argument (fallacies), such as the converse error or inverse error.

1.2. Predicates and Quantifiers:

1.2a. Translate between English sentences and symbols for universally and existentially quantified statements, including statements with multiple quantifiers. *

Sample Tasks:

- Translate a quantified statement into a logical expression, such as: “There exists an even prime number.”, “Every basketball player is tall.”, “For every positive rational number *a*, there exists a positive rational number *b* such that *ab=1.*”
- Make use of the symbols ∀ and ∃ when translating sentences from English into formal logic and vice-versa.

1.2b. Write the negation of a quantified statement involving either one or two quantifiers. *

Sample Tasks:

- Negate a universal statement, existential statement, and a statement involving two or more quantifiers.

1.2c. Determine if a quantified statement involving either one or two quantifiers is true or false.*

Sample Tasks:

- Determine if a given quantified statement, such as a quantified statement involving real numbers or integers and their properties, is true or false and justify the answer.

2. **Proof Techniques** - Successful discrete mathematics students understand how the rules of inference are used in standard proof techniques such as direct proofs, indirect proofs, proof by cases, and the use of counterexamples to disprove statements. They are able to apply these proof techniques to prove results from elementary number theory and other areas of mathematics.

The successful Discrete Mathematics student can:

2a. Use the direct proof method to prove propositions.*

Sample Tasks:

- Use the direct method to prove propositions about integers, such as *The product of an even integer and an odd integer is even.*
- Use the direct method to prove propositions about rational numbers, such as *The sum of two rational numbers is rational*.
- Use the direct method to prove propositions, such as *If x and y are positive real numbers, then* $\frac{x+y}{2} \geq \sqrt{xy}$.

2b. Identify logical errors and disprove statements. *

Sample Tasks:

- Explain the logical error(s) in a given incorrect “proof”.
- Provide counterexamples to disprove statements, such as *All real numbers y can be expressed as* $y = \frac{1}{x}$ *for some real number x*.

2c. Use mathematical induction to prove propositions. *

Sample Tasks:

- Let F_n be the n^{th} term of the Fibonacci sequence. Prove by induction that $F_1^2 + F_2^2 + \ldots + F_n^2 = F_nF_{n+1}$.
- Prove by induction that the sum of the first *n* positive integers is given by $\frac{n(n+1)}{2}$.

2d. Utilize other proof methods to prove propositions.

Sample Tasks:

- Use a proof by contradiction to prove propositions, such as $\sqrt{2}$ *is irrational.*
- Use a proof by contraposition to prove propositions, such as *If the product of two integers ab is even, then a is even or b is even*.
- Use a proof by cases to justify propositions, such as the *Triangle Inequality*.
- Write a constructive existence proof, such as *There exists a positive integer greater than 2 that can be written both as the sum of two cubes of positive integers and the sum of two squares of positive integers*.
- Write a nonconstructive existence proof, such as *There exist irrational numbers x and y such that* x^y *is rational.*
- Identify when a proposition is vacuously or trivially true.

3. **Set Theory** – Successful Discrete Mathematics students demonstrate an understanding of sets, set operations, set identities, and the cardinality of sets. They can use set notations including those for subsets, unions, intersections, differences, symmetric differences, complements, Cartesian products, the empty set, and power sets. They make use of Venn diagrams as appropriate, can formally verify set identities, and can apply the inclusion-exclusion principle to solve problems. They understand and use the terms cardinality, finite, countably infinite, and uncountably infinite, and can determine which of these characteristics is associated with a given set.

The successful Discrete Mathematics student can:

3a. Find subsets, unions, intersections, differences, symmetric differences, complements, power sets, and cross products of sets and use them to solve applied problems. *

Sample tasks:

- Determine if one set is a subset of another set.
- Find the union, intersection, and various cross products of two or more given sets, and use the idea of an empty set as appropriate.
- Find the difference of two given sets and the symmetric difference of those sets.
- Determine the complement of a given set from a specified universal set.
- Find the power set of a given set and determine the number of elements in the power set.

3b. Use Venn diagrams to solve problems, illustrate set identities, and apply the inclusion-exclusion principle. *

Sample tasks:

- Use Venn diagrams to demonstrate the ideas of subsets, unions, intersections, differences, symmetric differences, and complements of sets.
- Use Venn diagrams to illustrate various set identities, such as Domination Laws, Idempotent Laws, DeMorgan's Laws, and Absorption Laws.
- Use Venn diagrams to illustrate the principle of inclusion-exclusion for two or three sets.
- Apply the principle of inclusion-exclusion to determine the number of integers in a range of integers that are not divisible by 2 or 3.

4. **Relations** – Successful Discrete Mathematics students demonstrate an understanding of relations and are able to determine their properties. They can identify basic properties of relations, including the reflexive, symmetric, antisymmetric, and transitive properties. They can identify equivalence relations, equivalence classes, and partitions.

   The successful Discrete Mathematics student can:

   4a. Identify basic properties of relations, including reflexive, symmetric, antisymmetric, and transitive properties. Identify equivalence relations, equivalence classes, and partitions.

   Sample tasks:

   - Given the relation $R = \{(a, b) : a \leq b\}$, determine if the relation is reflexive, symmetric, antisymmetric, or transitive.
   - Determine if the relation R on students in a class where aRb if and only if a and b share a birthday (month and day) is an equivalence relation.
   - Show that the relation R on the set of all bit strings of length 5 where aRb if and only if they have the same bits in the first 3 positions is an equivalence relation; then determine the equivalence classes if so.
   - Determine if the relation R on the set of bit strings of length 4 where aRb if and only if they have exactly two bits in common is an equivalence relation. If it is, list the elements in the equivalence class [0110]; if it isn't, explain why.

   4b. Find the reflexive closure, symmetric closure, and transitive closure of a relation on a set.

   Sample tasks:

   - If R is the relation $a\mathrm{R}b$ if and only if a divides b on the set of integers, find the symmetric closure of R.
   - Find the smallest relation containing the relation {(1, 3), (2, 3), (4, 1), (4, 4)} that is reflexive and transitive.
   -

5. **Functions-** Successful Discrete Mathematics students demonstrate an understanding of functions and are able to determine their properties. They can determine the domain, codomain, and range of discrete functions. Given a function and a pre-image, they can determine its image, and given an image they can determine its pre-image(s). They can graph functions, perform composition of functions, find and/or graph the inverse of a function, and use the properties of functions to solve applied problems. They understand the notions of injections, surjections, and bijections, and can determine which of these characteristics are

associated with a given function. They can use the notion of a bijection to prove that two sets have the same cardinality. They can analyze the growth of elementary functions and determine their Big-*O* value. They can analyze simple algorithms and compare two algorithms based on computational complexity.

The successful Discrete Mathematics student can:

5a. Determine the domain, codomain, and range of discrete functions. Identify injections, surjections, and bijections, and determine which of these characteristics is associated with a given function. *

Sample tasks:

- Let f be the function from $\mathbb{Z}$ to $\mathbb{Z}$ given by $f(n) = n^2$. Determine the domain, codomain, and range of this function.
- Let f be the function from $\mathbb{Z}$ to $\mathbb{Z}$ given by $f(n) = n^2$. Determine whether this function is one-to-one and/or onto.
- Let f be the function from ($\mathbb{Z}$ x $\mathbb{Z}$) to $\mathbb{Z}$ given by $f(m, n) = m + n$. Determine whether this function is one-to-one and/or onto.
- Let f be the function from $\mathbb{Z}$ to $\mathbb{Z}$ given by $f(n) = \text{Floor}(1.5n)$. Determine whether this function is one-to-one and/or onto.
- Find a bijection from the even integers to the integers that are a multiple of 5 that proves these two sets have the same cardinality.

5b. Demonstrate an understanding of the terms cardinality, finite, infinite, and uncountably infinite, and determine the cardinality of a given set. Use the notion of a bijection to prove that two sets have the same cardinality.

Sample tasks:

- Determine whether the set of odd negative integers is finite, countably infinite, or uncountably infinite. If it is countably infinite, exhibit a one-to-one correspondence between the set of positive integers and the set of odd negative integers.
- Determine whether the set of real numbers between 0 and 3 is finite, countably infinite, or uncountably infinite. If it is countably infinite, exhibit a one-to-one correspondence between the set of positive integers and this set.
- Give an example of two uncountably infinite sets such that their intersection is countably infinite.

5c. Determine the Big-*O* estimate for basic functions.

Sample tasks:

- Show that $f(x) = x^3 + 7x^2 + 5x + 3$ is $O(x^3)$.
- Show that $g(x) = x^3$ is not $O(x^2)$.
- Determine an estimate for the big-*O* value of the sum of the first *n* positive integers.

6. **Sequences**- Successful Discrete Mathematics students demonstrate an understanding of sequence as a function whose domain is a subset of the integers. They can identify special sequences such as Fibonacci, factorial, arithmetic, and geometric sequences and can apply recursion.

The successful Discrete Mathematics student can:

6a. Define a sequence as a function whose domain is a subset of the integers. Identify arithmetic and geometric sequences, and the factorial sequence. *

Sample tasks:

- Find the first five terms of the sequence $\{a_n\}$, $n \geq 0$, where $a_n = 3(-2)^n + 5$.
- List the first ten terms of the arithmetic sequence whose first term is 5 and whose fourth term is 14.
- List the first 5 terms of the geometric sequence whose first term is 4! and whose third term is 96.

6b. Describe how sequences and sets can be defined recursively. Identify the Fibonacci sequence. *

Sample tasks:

- Give a recursive definition for the factorial sequence.
- Give a recursive definition for the Fibonacci sequence.
- Given $f(0) = 3$ and $f(n + 1) = 2^{f(n)}$, find $f(1)$ and $f(2)$.
- Given $f(1) = 5$, $f(2) = 2$, and $f(n+2) = 2f(n + 1) - f(n)$, find $f(3)$, $f(4)$, and $f(5)$.
- Let S be the set defined recursively by $3 \in S$, and if $x \in S$ and $y \in S$ then $x + y \in S$. Describe, in words, the set S.

7. Counting and Probability - A successful student will demonstrate understanding of the fundamental principles of counting and probability and the ability to apply them in a wide variety of situations.

The successful Discrete Mathematics student can:

7a. Solve counting problems involving the multiplication rule and permutations/combinations. (with and without repetition). *

Sample tasks:

- Use the multiplication rule to calculate the number of ways a process can be performed. For instance, suppose a computer installation requires one of four input/output units (A, B, C, and D) and one of three central processing units (X, Y, and Z). Determine how many different installations are possible.
- Compute the number of permutations of a set of n elements, for instance, the number of different ways you can arrange the letters in the word COMPUTER.

- Calculate the number of *r*-permutations of a set of *n* elements given by the formula $P(n,r) = \frac{n!}{(n-r)!}$.
- Use the binomial coefficient $C(n,r) = \binom{n}{r}$ to count the number of subsets of size *r* that are in a set of size *n.* For instance, find the number of five-member groups chosen from a set of twelve people.

7b. Apply the Addition Rule and the Principle of Inclusion and Exclusion. *

Sample tasks:

- Calculate the number of elements in a union of mutually disjoint finite sets.
- Calculate the number of elements in a union of sets when some of the sets have nonempty intersection.

7c. Apply basic principles of discrete probability.

Sample tasks:

- Calculate the probability of the union of events.
- Calculate the probability of the intersection of independent events, conditional probability, and probability of the intersection of dependent events.

7d. Investigate Pascal's Triangle and/or apply the Binominal Theorem.

Sample tasks:

- Given the n^{th} row of Pascal's Triangle students calculate the $(n + 1)^{th}$ row.
- Identify the relation between the binomial coefficients and the Pascal's Triangle entries.
- Use the binomial theorem to expand expressions such as $(1 + x)^6$.

7e. Apply the Pigeonhole Principle.

Sample tasks:

- Explain the pigeonhole principle.
- Apply the pigeonhole principle to problems, for instance, showing that any set of six distinct positive integers less than 11 contains a pair whose sum is 11.

8. Elementary Number Theory

The successful Discrete Mathematics student can:

8a. Determine if a proposed statement involving concepts from elementary number theory is true or false.

Sample tasks:

- Construct proofs involving concepts from elementary number theory such as properties of even and odd integers and divisibility.

- Determine if statements are true or false. Justify answers with a proof or a counterexample, as appropriate.

8b. State and use the Division Algorithm.

Sample tasks:

- State the Division Algorithm.
- Use the Division Algorithm to determine the equivalence relations in modular arithmetic.

8c. Apply modular Arithmetic.

Sample tasks:

- Perform additions, subtractions, and multiplications modulo *n*.

8d. State and use the Fundamental Theorem of Arithmetic.

Sample tasks:

- State the Fundamental Theorem of Arithmetic.
- Find the unique factorization of a given integer.
- Use the Fundamental Theorem of Arithmetic as a tool to find solutions to Diophantine equations.

8e. State and use the Euclidean Algorithm.

Sample tasks:

- State the Euclidean Algorithm.
- Use the Euclidean algorithm to hand-calculate the greatest common divisor of a pair of integers.

9. Graphs and Trees Theory - Successful Discrete Mathematics students understand the basic terminology used in graph theory and are able to use graphs and trees to model real-world situations. They are able to determine whether or not a given graph contains an Euler circuit/path and/or a Hamilton circuit/path, and construct those circuit(s)/path(s) if they exist. They are able to determine whether or not a graph is planar and apply Euler's formula if so. They understand the basic properties of *n*-ary trees, how to create decision trees, how to perform traversals, and use them to solve problems.

The successful Discrete Mathematics student can:

9a. Identify basic features of graphs, construct graphs with given properties, and represent graphs using matrices or lists.

Sample Tasks:

- Identify the neighbors of and the degree of each vertex in a graph.
- Identify the bridges in a given graph.

- Explain whether a given sequence of vertices determines a path in a graph, and if so, whether or not the path is simple and/or a circuit.
- Determine whether a given graph is simple or a multigraph, directed or undirected, connected or disconnected, and whether or not it is bipartite.
- Use a graph to model a real-world situation and explain how particular graph features align with the real-world situation.
- Create the complete graph K_n and the complete bipartite graph $K_{n,m}$ for given positive integers n, m.
- Represent a given graph with an adjacency matrix or list.
- Construct a graph represented by a given adjacency matrix or list.

9b. Determine whether or not a given graph has an Euler circuit, Euler path, Hamilton circuit, and/or Hamilton path and construct them if so.

Sample Tasks:

- Use Fleury's algorithm to find an Euler circuit/path in a given graph if one exists or explain why the graph does not have an Euler circuit/path.
- Determine the values of n, m for which K_n and $K_{n,m}$ have an Euler circuit or path.
- Identify a Hamilton circuit/path in a given graph if one exists or explain why the graph does not have a Hamilton circuit/path.

9c. Determine whether or not a given graph is planar and apply Euler's formula to planar graphs.

Sample Tasks:

- Determine whether or not a given graph is planar and draw a representation where no edges cross if so.
- Apply Euler's formula to make determinations about planar graphs with given characteristics.

9d. Identify trees and *n*-ary trees, create *n*-ary trees for specified applications including decision trees, and perform tree traversals.

Sample Tasks:

- Determine if a given graph is a tree, and if so, whether or not it is a *n*-ary tree.
- Create a decision tree to represent an algorithm, such as a sorting algorithm.
- Explain the process that is represented by a given decision tree.
- Construct a binary search tree for a list of strings.
- Perform tree traversals using preorder, inorder, and postorder traversal algorithms.
- Explain how traversals can be used to solve application problems.

10. Base-*n* Arithmetic- Successful discrete mathematics students understand base-*n* systems with a focus on base-2 (binary), base-8 (octal), and base-16 (hexadecimal) systems.

The successful Discrete Mathematics student can:

10a. Perform arithmetic in various base-*n* systems.

Sample Tasks:

- Count in binary, octal, and hexadecimal.
- Add, subtract, and multiply in binary, octal, and hexadecimal systems.

10b. Convert between various base-*n* systems.

Sample Tasks:

- Convert numbers between decimal, binary, octal, and hexadecimal systems.

10c. Represent signed binary numbers with 1 and 2's complements.

Sample Tasks:

- Represent signed binary numbers with 1 and 2's complements.
- Use 1 and 2's complements to add and subtract signed binary numbers.

11. Boolean Algebra/Logic - Successful discrete mathematics students understand the Boolean algebra structure and how it is related to logic networks. They are able to minimize Boolean algebra expressions and logic networks.

The successful Discrete Mathematics student can:

11a. Prove and/or apply properties of the Boolean algebra structure.

Sample Tasks:

- Determine whether a given mathematical structure is a Boolean algebra.
- Prove various properties of Boolean algebras.
- Construct a logic network to represent a Boolean expression.
- Write the truth function for a Boolean expression or logic network.
- Define a Boolean function and given a Boolean function, describe it using an input/output table.
- Write Boolean expressions in the canonical sum-of-products form given truth functions.

11b. Minimize Boolean algebra expressions and logic networks.

Sample Tasks:

- Minimize Boolean expressions and logic networks using Karnaugh maps and/or the Quine-McCluskey procedure.