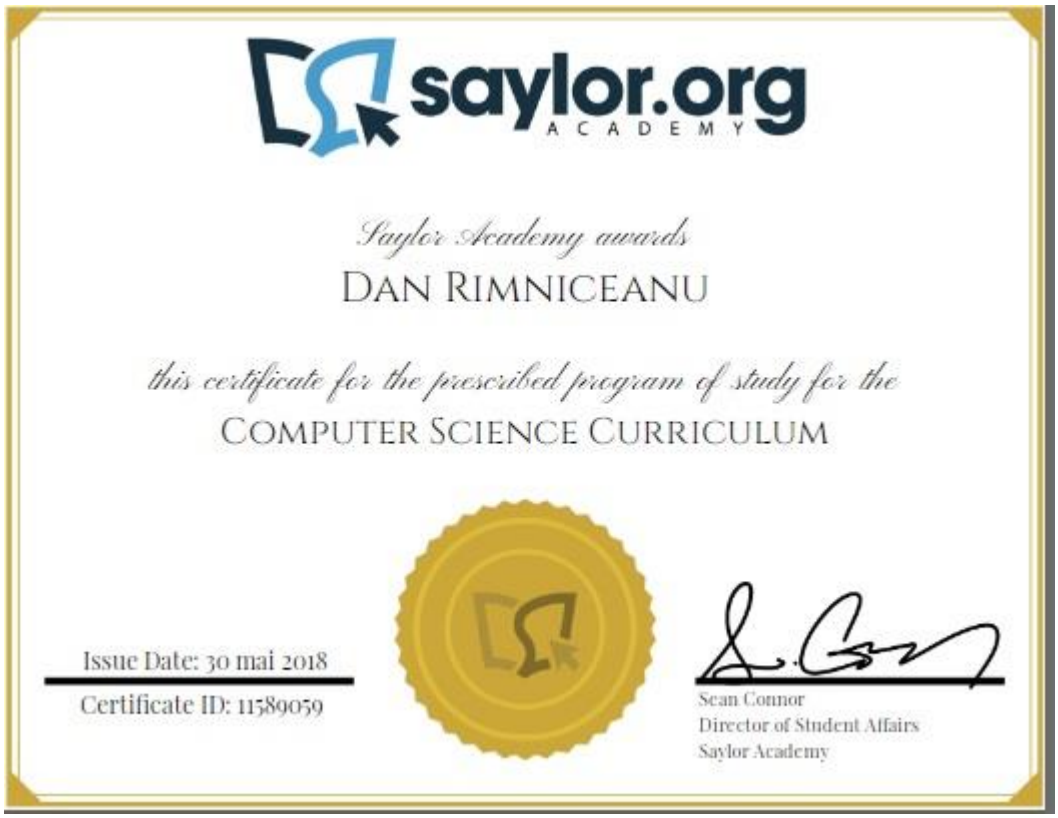
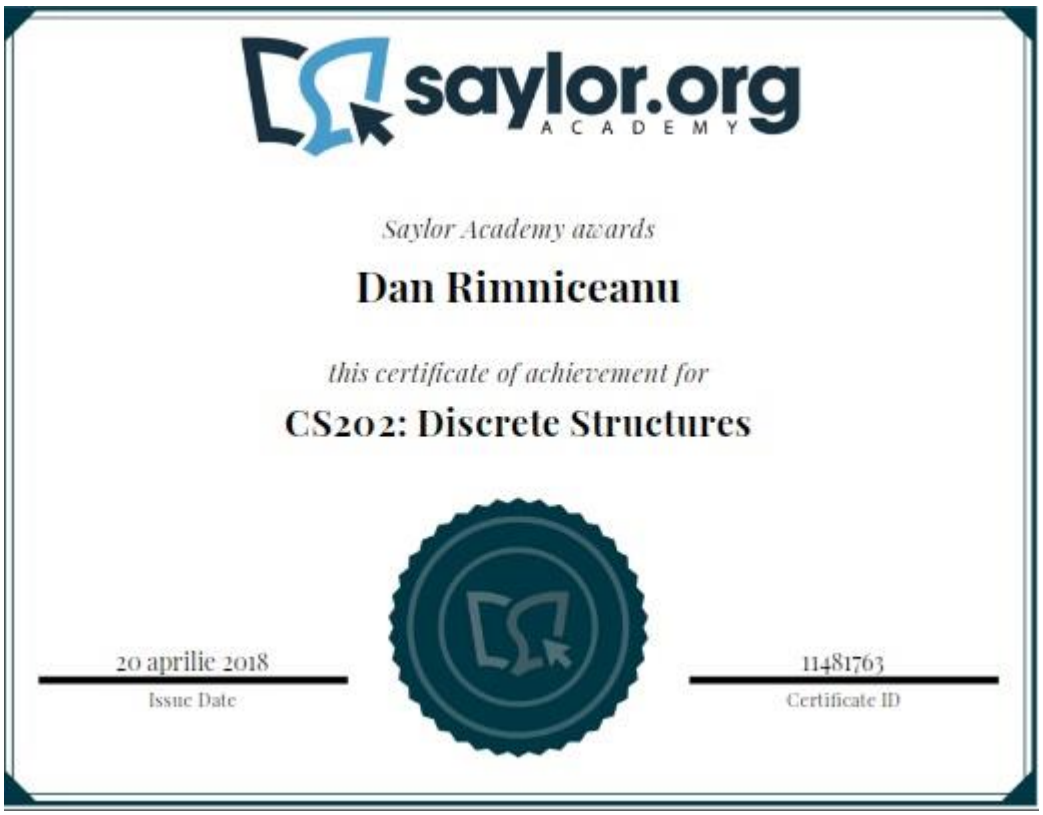

saylor.org
ACADEMY
Saylor Academy awards
DAN RIMNICEANU
this certificate for the prescribed program of study for the
COMPUTER SCIENCE CURRICULUM
Issue Date: 30 mai 2018
Certificate ID: 11589059
Sean Connor
Director of Student Affairs
Saylor Academy


saylor.org
ACADEMY
Saylor Academy awards
Dan Rimniceanu
this certificate of achievement for
CS202: Discrete Structures
20 aprilie 2018
Issue Date
11481763
Certificate ID

This provides a clear, accessible introduction to discrete mathematics that combines theory with practicality. Discrete mathematics describes processes that consist of a sequence of individual steps, as compared to forms of mathematics that describe processes that change in a continuous manner. The major topics we cover in this course are single-membership sets, mathematical logic, induction, and proofs. We will also discuss counting theory, probability, recursion, graphs, trees, and finite-state machines. Understanding the terms "single-membership" and "discrete" are important as you begin this course. "Single-Membership" refers to something that is grouped within only one set and systems that can be in only one state at a time, at the same hierarchical level. Similarly, "discrete" refers to that which is individually separate and distinct. Each of anything can be in only one set or one state at a time. This is a result of Aristotelian philosophy, which holds that there are only two values of membership, 0 or 1. An answer is either no or yes, false or true, 0% membership or 100% membership, entirely in a set or state, or entirely not. There are no shades of gray. This is much different from Fuzzy Logic (due to Lofti Zadeh), where something can be a member of any set or in any state to some degree or another. Degrees of membership are measured in percentage and those percentages add to 100%. But, even in Fuzzy Logic (multiple-membership, multiple-state, non-discrete logic), one ultimately comes to a crisp decision so that some specific action is taken, or not. For this course, it is enough to understand the difference between single-state and multi-state logic.

As you progress through the units of this course, you will develop the mathematical foundation necessary for more specialized subjects in computer science, including data structures, algorithms, cryptology, and compiler design. Upon completion of this course, you will have the mathematical know-how required for an in-depth study of the science and technology that is foundational to the computer age.

Unit 1: Sets, Set Relations, and Set Functions

Computer scientists often find themselves working with groups of homogeneous or heterogeneous entities. Mathematicians devised single-membership set theory to respond to these situations. In this unit, we will cover the theoretical background of sets and take a look at associated definitions, notations, relations, and functions. This is a fundamental tool of mathematics and computer science, and is essential to understanding the other topics in this course.

Completing this unit should take you approximately 5 hours.

- Upon successful completion of this unit, you will be able to:
    - define sets, operations on sets, and state important set properties;
    - categorize sets into their various types (such as singleton, finite, infinite, equal, null, proper subset, and improper subset) and give a definition of each;
    - describe set notation and how that notation is used to perform operations via symbol manipulation; and

- apply set definitions, operations, and properties to demonstrate set membership within a specific context.

- 1.1: Set Notation and Relations
  - Set Notation and RelationsPage
    Every field of study seeks a common terminology and symbology. While it is possible to think about a subject without knowing its particular language, it is not possible to communicate with others about that subject without some common frame of reference. Thus we begin with the basic terms and notations of set theory.
  - NotationPage
    This table defines the notation used in this course.
  - Try It NowBook
    Work these exercises to see how well you understand this material. Hints and solutions to some of these exercises are given on the second page. As the course proceeds, new notations and terminology will be introduced as they are needed. Be sure to not plug-and-chug answers to the exercises. These are to aid your understanding, and it will be difficult to pass the exam if you do not take the exercises seriously. While you should not copy others' work, feel free to discuss these exercises with others who are taking this course.
- 1.2: Basic Set Operations
  - Basic Set OperationsPage
    Even as set members are discrete, so are sets themselves. The question we ask about each member is, "Of what sets is it entirely a member?" Although there are no partial set memberships, an entity can be entirely a member of more than one set. So, we can perform various operations on sets, such as add one set to another and subtract one set from another. With questions that require a Yes or No response, there is no dual membership since those are on the same hierarchical level. However, with layered hierarchies, dual 100% membership is possible. For example, we can talk about a car that is also a vehicle. The entity exists entirely in two different sets. Later in the course, we will talk more about hierarchies.
  - Types of SetsPage
    Read this page to become familiar with the various types of sets. This page is an aid to set terminology and notation.
  - Try It NowBook
    Work these exercises to see how well you understand this material.

- 1.3: Cartesian Products and Power Sets
    - Cartesian Products and Power SetsPage
      While the last section discussed combining sets of individual members to create another set of individual members, here we discuss creating sets of non-repeating tuples (pairs, triplets, and higher groupings). Later in the course, we will see how to calculate the number of tuples that would be created under various circumstances.
    - Try It NowBook
      Work these exercises to see how well you understand this material.
- 1.4: Summation Notation and Generalizations
    - Summation Notation and GeneralizationsPage
      We have dealt with relatively straightforward notation so far. However, as situations involving sets become more complex, a more compact notation is needed. Here presented is an introduction to that notation.
    - Try It NowBook
      Work these exercises to see how well you understand this material.

Unit 2: Counting Theory

"How many elements do we expect to find within a given set?" This is a simple question, but hard to answer correctly without guessing. Counting theory (also known as combinatorics) lets us do that through several different approaches that depend on the circumstances of the given problem. These approaches are not difficult, but you have to know when to use each one, and which circumstances each approach applies to. In this unit, we will carefully walk through these considerations.

Completing this unit should take you approximately 5 hours.

- Upon successful completion of this unit, you will be able to:
    - describe counting, permutation, and combination rules;
    - compute the number of permutations and combinations of the members of a given set; and
    - determine the number of all possible outcomes of a collection of events.

- 2.1: Rule of Products
    - Introduction to the Rule of ProductsPage
      If there is some number of different operations, and each operation can be performed in different ways, how many combinations of operations can be performed when the operations are not dependent on one another? This is an

important question when trying to determine the number of outcomes for which one must account.

- Try It NowBook

Work these exercises to see how well you understand this material.

- 2.2: Permutations
  - Ordering Elements of a SetPage

Often it is the case that the order of things is not important. That is where permutations come in. It enables us to show how many different arrangements can be had of a given set of elements. Of course, this is not true of something like an event stream but we will discuss those later.

  - Try It NowBook

Work these exercises to see how well you understand this material.

- 2.3: Partitions of Sets and the Law of Addition
  - Partitions and Addition LawsPage

In how many ways can a set be partitioned, broken into subsets, while assuming the independence of elements and ensuring that each element appears in only one subset? This question is often answered, for instance, when deciding how to partition workload across a network of computers. As systems become increasingly complex, the answer allows us to make best use of available resources.

  - Try It NowBook

Work these exercises to see how well you understand this material.

- 2.4: Combinations and the Binomial Theorem
  - Combinations and the Binomial TheoremPage

We now consider subsets of a given size. How many subsets of a given size can be created from the elements of a given set when order is not a concern? This is a good question when deciding how to partition supply to meet demand when a given number of units are required, without specifying which units.

  - Try It NowBook

Work these exercises to see how well you understand this material.

Unit 3: Mathematical Logic

In this unit, we will discuss basic concepts that are part of the foundation of mathematical logic. As you seek to fully understand these concepts, you must be able to recognize valid logical arguments. Although these arguments will usually be applied to mathematics, they are the same techniques used by lawyers in the courtroom, physicians examining a patient, or engineers trying to solve a difficult problem. The circuits of computers are designed using the same algebra of propositions that we will discuss in this unit. Often, embedded computers (that is, the data processing units within a larger machine) are programmed by using binary, gate-level logic, machine code, or ladder logic. These rely on the basic concepts we will discuss here.

Completing this unit should take you approximately 5 hours.

- ### 3.1: Propositions and Logical Operators

  - Propositional LogicPage

    Within this subunit, we encounter basic definitions and operators. Fundamental symbology is also presented and discussed. There are several examples that help you understand the math in terms of human language. You will see several ways to say the same thing, while remaining logically and mathematically correct.

  - Propositions and Logical OperatorsPage

    Read these sections to supplement your understanding of propositional logic.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- ### 3.2: Truth Tables and Propositions Generated by a Set

  - Summary of Logic NotationPage

    Be sure to review this notation summary since these terms will be used throughout this unit.

  - Truth Tables and Propositions Generated by a SetPage

    What is in a set and what is not in a set leads to some interesting ways of analyzing truth or falsehood. In this section we use 0 for false (no) and 1 for true (yes). One can also speak in terms of "do-not" or "do", "do not perform this action" or "do this action". It is a matter of interpretation, an interpretation that must be established and remain consistent. We can write equations to express these ideas so that many factors can be considered and operated upon in a standard way. This section starts you down that path.

- Try It NowBook

  Work these exercises to see how well you understand this material.

- 3.3: Equivalence and Implication
  - Equivalence and ImplicationPage

    There are many ways to write the same logical equation. Too, various equations are implied by other equations or are contradicted. This section explores that idea. For instance, changing the terminology used to describe an idea, object, or field of study does not change those from what they are. An example are cloud providers who reinvent basic terminology in distributed systems so that their offering appears to be new and unique. An understanding of this area of thought will allow you to see through marketing hype and to simplify logical equations. Thus, you can bring clarity to your understanding and lower costs to systems described by logical expressions.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- 3.4: The Laws of Logic
  - The Laws of LogicPage

    We will now prepare for the unit on proofs. Essentially, a table of laws is presented and discussed. These are essential to our future study in this topic area. You will find a similarity between laws of logic and laws of algebra. However, just as similarities between the syntax of computer languages can lead you astray, be sure you keep logic and algebra separate. For instance, 1 + 1 does not equal 2 in logic. Rather, 1 + 1 = 1.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

Unit 4: Mathematical Induction and Proofs

So far, we have learned about symbology, truth tables, writing equations, and determining how many units could be in a set under specific circumstances. We have done all of this has under the assumption that we can easily see the results of our efforts. However, in practical situations that you will encounter in real life with serious applications, we cannot always see our results quickly, or necessarily even traverse the situation easily. For that, we must turn to the methods behind mathematical systems and proofs. These allow us to "get the big picture" without having to visualize it all at once. Proofs also give us a way to consider all of the available information in a given

arrangement of facts, even when some of the "facts" might not actually be true. They also have the added advantage of letting us ignore whole portions of a situation when we know they do not hold the answer we seek.

Completing this unit should take you approximately 5 hours.

- Upon successful completion of this unit, you will be able to:
  - discuss how one conclusion leads to another, in the mathematical sense, for a given situation;
  - write the terms of a logical sequence as a generalized formula;
  - use mathematical induction to prove the validity of logical sequences; and
  - employ inductive reasoning to situations where there is sufficient evidence to warrant generalization.

- 4.1: Mathematical Systems
  - Mathematical SystemsPage

    This page gives an overview of what a mathematical system is and how logic plays an important role in such systems. A discussion on how mathematical facts are developed and organized will help to unify the concepts presented later in the course. The system of propositions and logical operators we developed earlier will serve as a model for our discussion.

- 4.2: Direct Proof
  - Direct ProofPage

    "Brute force" is a term commonly applied to considering all possibilities manually, one at a time, before coming to a conclusion. That works well under simple circumstances but is rarely practical within industrial applications. This section discusses that reality so that you can avoid the "brute force" trap.

- 4.3: Indirect Proof
  - Indirect ProofPage

    Science works to either prove or disprove assertions. This is contrary to those who insist that science seeks only to disprove assertions. This mentality causes the acceptance of assertions unless they are proven false. Therein lies a dangerous way of thinking since it leads to "guilty until proven innocent" once an accusation is made. Assertions are not acceptable as fact until they are proven true. In this section, we consider "proof by contradiction", one side of scientific effort.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- 4.4: Propositions over a Universe
    - Propositions and Truth SetsPage

      As one enters into the realm of advanced technology, one increasingly realizes that a layman's use of terminology and language is too imprecise. Not only does the use of technology require precision, but so does its discussion. Otherwise, it is not possible to establish requirements for a project. Nor is it possible to discover the cause of system problems and thereby get a sense of where to focus one's energies. This subunit gets you thinking along those lines and helps you to understand and avoid the vagueness of common speech.

    - Try It NowBook

      Work these exercises to see how well you understand this material.

- 4.5: Mathematical Induction
    - Discussion and ExamplesPage

      Induction is a means of proving a theorem by showing that if the theorem or assertion is true of any particular case, it is true of the next case in a series, and then showing it is indeed true in any particular case. Our text applies that definition to mathematics by saying mathematical induction is a technique for proving propositions over the positive integers. Mathematical induction reduces to a finite number of steps the proof that all positive integers belong to a specific truth set. The number of steps to complete the proof is the same, regardless of the size of the set or the number of positive integers involved. This is crucial to our discussion of discrete mathematics since it allows us to consider large volumes of evidence while deciding whether to accept or reject an assertion. As we develop automated systems or simply consider what we are hearing on the evening news, we find ourselves having to do exactly that.

- 4.6: Strong Induction
    - Strong Mathematical InductionPage

      Also known as course-of-values induction, strong induction adds to mathematical induction by showing that additional assertions hold if mathematical induction holds.

    - Try It NowBook

      Work these exercises to see how well you understand this material.

Unit 5: Probability

Probability is an important component of discrete mathematics. For instance, if you consider a population of 10 and then take four at a time, without duplicates, what is the chance that a certain combination within the subsets of four will occur? Or, given the set of all possible events, what is the chance that a certain event will occur, given that the event can result from various causes? The construction of trees and graphs , which we will discuss later, depends on the probability of combinations of events. Since we want to traverse trees and graphs in order of the most likely and most important events, we need to know probability.

Completing this unit should take you approximately 4 hours.

- Upon successful completion of this unit, you will be able to:
  - state the definitions of terms that apply to probability within the context of this course;
  - calculate conditional probabilities (the probability of event Y occurring if event X has already occurred);
  - compute the probability of independent events; and
  - estimate the chance of occurrence of a specific event within a collection of events.

- 5.1: Introduction
  - Introduction to Discrete ProbabilityPage

    We'll begin by laying the foundation of your study of probability. This video will walk through basic concepts you will need to know.

- 5.2: Sample Spaces, Events, and Their Probabilities
  - Sample Spaces, Events, and Their ProbabilitiesPage

    Next, we learn about the sample spaces associated with random experiments, about the events that can occur from random experiments, and have a look at a specific event's probability of occurrence.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- 5.3: Complements, Intersections, and Unions
  - Complements, Intersections, and UnionsPage

    There are situations where a problem can be solved by representing it as a simpler problem, or one that is similar and already solved. We discuss those opportunities here.

- Try It NowBook

  Work these exercises to see how well you understand this material.

- 5.4: Conditional Probability and Independent Events
  - Conditional ProbabilityPage

    Events can be dependent or independent of something else. This video considers events whose probability of occurrence is independent of all other events.

  - Independent EventsPage

    Read this section.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

Unit 6: Recursion

Using recursion, we can arrive at a succession of elements (such as numbers or events) by operating on one or more of the elements that came earlier. We do this using a rule or formula with a finite number of steps. In computer programming, we implement recursion with procedures, subroutines, functions, or algorithms that run one or more times until a specified condition is met. At that point, the remaining code in each procedure is processed from the last procedure called to the first. From a practitioner's perspective, recursive procedures are simple to write, but they are extremely memory-intensive, and it can be difficult to predict how much memory will be required.

Recursion is common, so you will need to understand it at a fundamental level. A basic example of a recursive sequence is Dt = f(D[t-1]). The data at time t is a function of the data at the previous unit of time. In practice, this can be implemented as a recursive function or an explicit (that is, non-recursive) function. This unit will delve deeper into this topic.

Completing this unit should take you approximately 5 hours.

- Upon successful completion of this unit, you will be able to:
  - define linear recurrence and give examples;
  - explain important recursive functions;
  - produce recursively defined sequences; and
  - write recursive relations that are defined in terms of themselves at increasingly-earlier moments in time.

- 6.1: Introduction

- Recursion DemystifiedURL

  Recursion, something being defined in terms of a different version of itself, can be somewhat mystifying, so read this section for a gentle introduction before we move on to something more formal.

- Recursive DefinitionsPage

  Read this page to supplement to your understanding of recursion.

- 6.2: Transitioning from Informal to Formal
  - Collected Definitions of RecursionPage

    Now, we will take a look at recursion from informal and formal perspectives that are related. Illustrations give us a practical sense of the matter. This approach prepares us for an increasingly-formal discussion. Read this page to see how recursion exists beyond mathematics and computer programming.

  - Explanation and ExamplesPage

    Read this page to get a sense of recursion from a mathematical basis.

  - Illustrating RecursionPage

    This video explains recursion via illustration.

  - Sequences and RecursionPage

    This video relates numeric sequences to recursion. There is also a discussion on notation.

- 6.3: The Many Faces of Recursion
  - The Many Faces of RecursionPage

    There are many ways to approach recursion. We take a look at those here.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- 6.4: Sequences
  - SequencesPage

    In this subunit, we delve into progressions of numbers and their subsets. These progressions can be defined by recursive processes.

- Try It NowBook

  Work these exercises to see how well you understand this material.

- 6.5: Recursion in the Real World
  - Predicting Data Sample ValuesPage

    Read this example of a practical, complex, real-world problem that uses recursion in many ways. The underlying math is not testable, but try to get a sense of its subtle recursive nature.

Unit 7: Graphs

Graphs are formal mathematical representations of networks, collections of objects, events, or set elements that lead naturally from one to another. In this unit, we will examine the formality of graphs. Graphs are extremely useful in business, science, and engineering. In this unit, we will discuss how to understand graphs, how to build them, how to manipulate them, and how to use them.

Completing this unit should take you approximately 9 hours.

- Upon successful completion of this unit, you will be able to:
  - state the components of a graph;
  - describe the two principal graph traversal paradigms;
  - explain the difference between directed and undirected graph; and
  - use graphs to solve applications involving associated events.

- 7.1: A Less-Formal Introduction
  - Basic Graph TheoryPage

    We begin with a top-level view of graphs by looking at definitions, structure, and dynamics.

- 7.2: A Formal Introduction
  - Introduction to GraphsPage

    Now that you have the general idea, we proceed with the formality required to gain an intuitive idea of what graphs are and what could potentially be done with them.

  - Try It NowBook

    Work these exercises to see how well you understand this material.

- 7.3: Graph Structures
    - Data Structures for GraphsPage

      Transitioning to practicality, we now learn about how data can be structured so that graphs are correctly formed for application.

    - Try It NowBook

      Work these exercises to see how well you understand this material.
- 7.4: Graph Node Connectivity
    - ConnectivityPage

      Which nodes are connected, directly or indirectly? Between which nodes is there a path from one node to another? Those are the questions we deal with now. This is part of the topic of graph traversal, encountering a series of connected nodes in a graph or subgraph.

    - Try It NowBook

      Work these exercises to see how well you understand this material.
- 7.5: Graph Traversals
    - Traversals: Eulerian and Hamiltonian GraphsPage

      At times, we need a way to encounter every node in a graph. This page discusses how to do just that by using two important kinds of graphs.

    - Try It NowBook

      Work these exercises to see how well you understand this material.
- 7.6: Graph Optimization
    - Graph OptimizationPage

      A common thread that connects all the topics in this unit is the opportunity to optimize (maximize or minimize) some quantity that is associated with a graph. For instance, we may decide to select a path from one node to another based on the weight of the connections between those nodes. The weights refer to some aspect of the situation at hand, like distance, cost, or importance.

    - Try It NowBook

      Work these exercises to see how well you understand this material.

- 7.7: Planarity and Colorings
    - Planarity and ColoringsPage

      Annotating a graph so that it contains additional information is very important as the complexity of an application increases. Generally, the more complex the technology, the more precise needs to be its descriptive information.

    - Try It NowBook

      Work these exercises to see how well you understand this material.

Unit 8: Trees

Trees are a special case of graphs. Certain assumptions are made about the structure of trees, so various actions can be taken with them that would not work with graphs. Put simply, a tree is a nondirected graph where any two nodes are connected by exactly one path. There are no cycles in a tree. Although this definition is very simple, it has powerful consequences within graph theory and in the real world. In this unit, we will explore trees further.

Completing this unit should take you approximately 4 hours.

- Upon successful completion of this unit, you should be able to:
    - define the basic components of a tree;
    - explain trees as a special case of graphs;
    - discriminate between binary and nonbinary trees; and
    - identify subgroups within a specific graph that includes all vertices and a minimum number of edges.

- 8.1: Introduction
    - Quick Introduction to TreesPage

      Read this page. Take careful note of its definitions.

    - Formal IntroductionPage

      Getting a good grasp of the definition of a tree and of its components is essential to understanding and making good use of them.

    - Try It NowBook

      Work these exercises to see how well you understand this material.

- 8.2: Spanning Trees
    - OverviewPage

Read this page to get a basic overview of spanning trees.

- Detailed DiscussionPage

A spanning tree T of an nondirected graph G is a subgraph that itself is a tree that includes all vertices of G, with a minimum possible number of edges. In general, a graph may have several spanning trees. This concept has a lot of implications regarding optimization of tree structure and traversal. Let us have a look at this idea.

- Try It NowBook

Work these exercises to see how well you understand this material.

- 8.3: Rooted Trees
  - Rooted TreesPage

We can separate rooted trees from nondirected trees by noting that a rooted tree contains a special vertex, called the root. If we choose any other vertex in the tree, we know that there is a unique path from the root to that vertex. The implication is that there is a hierarchy of vertices. Lets us take a deep look at rooted trees.

  - Try It NowBook

Work these exercises to see how well you understand this material.

- 8.4: Binary Trees
  - Binary TreesPage

Binary trees are special cases of rooted trees. Binary trees have zero or more nodes. Each node has at most two children. A node without children is called a leaf. This type of tree is the most commonly encountered in practice. That is why we spend additional time with them.

  - Try It NowBook

Work these exercises to see how well you understand this material.

Unit 9: Finite-State Automata

A finite-state machine (FSM) is a mathematical model of computation that describes an abstract machine in one of a finite number of states at any point in time. The FSM can change from one state to another as it responds to data inputs, or when some condition is satisfied. The change from one state to another is called a transition. An FSM is defined by a list of its states, its initial state, and the conditions for each transition. Often, state

machines are illustrated as graphs whose nodes are the states and whose links are the transition conditions.

Completing this unit should take you approximately 2 hours.

- Upon successful completion of this unit, you will be able to:
  - illustrate abstract machines that can be in exactly one of a finite number of states at any given time;
  - analyze systems that recognize input patterns, accepting or rejecting an input depending on whether a given pattern occurs;
  - construct discrete-state systems; and
  - predict how a given system will enter different states as new data is input to the system over time.

- 9.1: Introduction
  - Finite-State Machine OverviewPage

    Read this article for a quick look at finite state machines.
  - More on Finite State MachinesPage

    This short video explains further.
  - State MachinesPage

    Read this section to go into more detail on FSMs.
- 9.2: State Transition Diagrams
  - State Transition DiagramsPage

    How we illustrate a finite-state machine has a great deal to do with how well others will understand our design. There is also an opportunity for objective review and evolution of the underlying system. This video continues with a discussion on the design of a combination lock.
- 9.3: Finite-State Machine States
  - FSM StatesPage

    This short video explains more about states in a finite-state machine.
  - FSM ExamplesPage

    Review these example designs of relatively simple practical applications of finite-state machines.

- 9.4: Putting the Basics to Use
    - Putting the Basics to UsePage

      Watch these videos for examples of using FSMs in the real world. They also discuss some subtle issues related to implementing FSMs.