Preview: MAT2051 : Discrete Mathematics

Syllabus

Course Overview

Course Introduction

Launch Presentation | Transcript

Although a fairly new discipline, discrete mathematics has become an important requirement for the fields of computer science, computer information systems, and business management. The reason for this is that discrete mathematics provides the mathematical tools necessary for programming computer algorithms and calculating combinations and probabilities of strategic alternatives. It also introduces graphing techniques for both business and computer science and uses graphs to analyze problems and help make decisions.

The course uses a combination of activities to help you build a solid framework in discrete mathematics and to introduce practical applications of its techniques and algorithms. These include textbook readings, assignments, discussions, and quizzes.

Course Competencies

(Read Only)

To successfully complete this course, you will be expected to:

1. Describe how discrete math theories are used in computer science.
2. Apply the methodologies of discrete math.
3. Use mathematical reasoning, deductive and inductive logic, and proofs to solve problems.

4 Apply discrete math methods and tools to solve problems encountered in a work setting.

Course Prerequisites

Prerequisite(s): MAT1050 or MAT-FP1050.

T

Syllabus >> Course Materials

Required

The materials listed below are required to complete the learning activities in this course.

Integrated Materials

Many of your required books are available via the VitalSource Bookshelf link in the courseroom, located in your Course Tools. Registered learners in a Resource Kit program can access these materials using the courseroom link on the Friday before the course start date. Some materials are available only in hard-copy format or by using an access code. For these materials, you will receive an email with further instructions for access. Visit the Course Materials page on Campus for more information.

Book

Johnsonbaugh, R. (2018). *Discrete mathematics* (8th ed.). New York, NY: Pearson. ISBN: 9780321964687.

External Resource

Please note that URLs change frequently. While the URLs were current when this course was designed, some may no longer be valid. If you cannot access a specific link, contact your instructor for an alternative URL. Permissions for the following links have been either granted or deemed appropriate for educational use at the time of course publication.

- Purdue University. (2010). *Paraphrase: Write it in your own words.* Retrieved from http://owl.english.purdue.edu/owl/resource/619/01/

Suggested

The following materials are recommended to provide you with a better understanding of the topics in this course. These materials are not required to complete the course, but they are aligned to course activities and assessments and are highly recommended for your use.

Optional

The following optional materials are offered to provide you with a better understanding of the topics in this course. These materials are not required to complete the course.

External Resource

Please note that URLs change frequently. While the URLs were current when this course was designed, some may no longer be valid. If you cannot access a specific link, contact your instructor for an alternative URL. Permissions for the following links have been either granted or deemed appropriate for educational use at the time of course publication.

- Martin, D. R. (2007). *Insertion sort.* Retrieved from http://www.sorting-algorithms.com/insertion-sort

Unit 1 >> Set Theory, Logic, and Logic Circuits

Introduction

"Logic is the beginning of wisdom . . . not the end."

— Star Trek VI:
The Undiscovered Country

All fields of study have a unique language through which those who study and work in that field can communicate. The language of mathematics, through its exquisite elegance and aim to disambiguate, is perhaps the most commonly spoken language in the world. As we progress through this course, the textbook, and applications of technology and computer science, the many subsets that make up but a fraction of the language of mathematics will be introduced and applied.

Sets

Chapter 1 introduces set theory and logic. A set is simply a collection of objects with order not taken into account. A graph, for example, is a set of edges and vertices. Our number system is made up of sets, some containing others. The real numbers are an infinite set of all values between negative infinity and positive infinity. The set of real number contains a smaller infinite set of the integers. Sets can be used to describe groups and objects, relationships, containment, equivalence, and so on.

Logic and Circuits

Logic is sometimes known as the study of reasoning, or determining, the truth or correctness of given statements or arguments. Logic statements can use any language, such as algebra or English.

The following are logic statements:

- $x + 2 = 3$.
- Anyone who wears socks and sandals has a PhD.

In Chapter 11, you will examine an application of sets and logic known as logical circuits. Logical circuits used in computers have one of two outputs: 0 or 1. Computer systems and all related applications are built using this seemingly simple on-or-off option. As we explore Chapter 11, we will look at logic tables, gates, and related properties.

The best first step in any math class is to open the textbook and start reading. The examples will be your first tutorials, and we will work together throughout this course to learn about the mathematics underlying computer science and information technology.

Reference

Meyer, N., & Flinn, D. M. (1991). *Star Trek VI: The undiscovered country*. Retrieved from http://thinkexist.com/quotes/leonard_nimoy/

Learning Activities

u01s1 - Studies

Introductions

The unit introductions contain important information that will help you throughout this course. In some instances, there are illustrations to help you visualize topics addressed in the unit. It is an expectation

that you read through the introduction before starting work in its particular unit. This is the expectation for each unit of the course.

Multimedia

In Units 1 through 9 you will be learning a variety of concepts that make up discrete mathematics. To help you learn these concepts in detail, videos cover the topics presented in that particular unit. This unit's presentation is called *Logic*. It includes the following topics:

- Definitions.
- Truth table of compound statements.
- Conditional propositions.
- Logical equivalence.
- Tautology.
- De Morgan's law.

Click **Logic** to begin the presentation.

Readings

Read the following in your *Discrete Mathematics* text:

- Chapter 1, "Sets and Logic," sections 1.1 through 1.6, pages 1–53. These sections address sets, propositions, conditional propositions, logical equivalence, arguments, and rules of inference.
- Chapter 11, "Boolean Algebras and Combinatorial Circuits," sections 11.1 and 11.2, pages 532–542. These sections focus on combinational circuits and their properties.

u01s2 - Practice Problem Set

In Units 1–9, you will have an opportunity to work on practice problems to help you with the concepts for that unit. This will take the following format:

1. A set of problems to solve from the book.
2. An interactive presentation that shows the steps of solving each problem and provides the answers.
3. A discussion for you to post and discuss these problems—and their solutions—with your peers.

These problems will prepare you for the quizzes throughout the course and the culminative final exam in Unit 10. Practice problem sets are not graded and will not be collected. You are, however, encouraged to complete them, as they will help you prepare for the quizzes.

The quiz in Unit 2 will cover all topics in Units 1 and 2.

Practice Problems for This Unit

Before viewing the interactive presentation that follows, see how you do on your own with the following problems from the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 1.1, page 12, exercises 34, 45, and 67.
- Section 1.2, page 19, exercises 25, 28, and 49.
- Section 1.3, page 30, exercises 60 and 78.
- Section 1.4, pages 35–36, exercises 14, 31, and 32.
- Section 11.1, page 537, exercises 4 and 9.
- Section 11.2, page 543, exercises 2 and 8.

Multimedia

Click **Practice Problem Sets for Unit 1** to learn a step-by-step process for solving each problem for this set.

u01s3 - The Ethical Use of Intellectual Property at Capella

Plagiarism and Academic Honesty

In our modern world, we have nearly unlimited access to others' thoughts, ideas, and written works via the Internet and other search tools. Because many of us exchange ideas both in our work and in our creative lives, it has become increasingly important for us to practice ethical uses of intellectual property.

Throughout your Capella career, you will conduct research, evaluate sources, and interpret and synthesize the work of others. Your final product—be it an essay, discussion post, or final project—

should integrate your own voice (that is, your original ideas, interpretation, or written analysis) with properly cited ideas, theories, and concepts from others. Failure to properly credit your sources is considered plagiarism and is a violation of the Capella University Academic Honesty Policy.

Plagiarism is a type of academic dishonesty that we take seriously. When you engage in this behavior, you put your academic career at risk. Understanding plagiarism and how to avoid it is important for any learner in Capella courses. This activity addresses the issue of plagiarism.

Read "Plagiarism: What It Is and How to Avoid It." The following resources provide additional insight on plagiarism and academic honesty:

- Capella Online Writing Center.
- Capella Online Writing Center – Academic Integrity.
- Capella Online Writing Center – Academic Integrity and Plagiarism Handout.
- Capella Online Writing Center – Plagiarism Handout.
- Capella University – The Writing Process and Academic Honesty.

Paraphrasing

Paraphrasing is an important skill to develop. It is how you read or hear another author's ideas, critically evaluate and synthesize them with your own ideas, and restate them in your own words. An academic paper should typically contain elements of original work (your thoughts), the work of supporting authors (via direct quotes with formal APA citations), and paraphrasing (where you synthesize your thoughts with those of others and draw unique, educated connections on a given topic or question). You will spend much of your academic career developing and honing this skill.

Review the Purdue Online Writing Lab's *Paraphrase: Write It in Your Own Words* Web page to help you push your writing to the next level, academically speaking.

Resources and Tools

Visit the Capella Online Writing Center's APA Citing and Referencing page for additional APA guidance, including examples of citations for a variety of sources.

Visit the Capella University Library – RefWorks page for more information and webinars on using RefWorks to help organize and prepare your sources.

Visit the SafeAssign Campus page for information and tutorials on this helpful tool. SafeAssign is a tool you can use to ensure the integrity of your texts when using researched, published material. Please make note of the various ways that you can use SafeAssign in the future.

- Use the SafeAssign Draft option to check your writing and ensure that you have paraphrased, quoted, and cited your sources appropriately. Run a SafeAssign report, saving your paper as a

draft.

- Based on your SafeAssign results, make any necessary changes to your paper before submitting your final assignment to your instructor for grading.

u01d1 - Logic and Reasoning

Logic and reasoning play important roles in computer applications, systems management, database management, and error management. In this discussion, you will examine logic and deductive reasoning, symbols and definitions, and an application involving database and error logging.

Deductive Reasoning, Part 1

Given the following true logical statement:

If inputs x_1, x_2, and x_3 are all true, then output y is true.

1. Write this statement as a logical statement using the appropriate logic symbols for conjunction and implication.
2. Write the converse and contrapositive of this statement.
3. Is the converse true? Explain why or why not.
4. Is the contrapositive true? Explain why or why not.

Deductive Reasoning, Part 2

Assume you are a systems manager and that you know a server will log an error message if a database has a retrieval error. With that, complete the following:

1. Why is it important for errors to be logged?
2. What is a database retrieval error, and what might cause this type of error?
3. If you notice that there is an error logged, can you infer that a retrieval error has occurred without reading the log? Explain why or why not. (Hint: This is a question about reasoning.)

Prior to posting your response, review the Discussion Participation Scoring Guide. This is the recommendation for all discussions in this course.

Response Guidelines

Read your peers' posts and respond to two. Do you agree with your peer's assessment of the problem? Explain.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u01d2 - Practice Problem Set Review

The problem sets are assigned to help you practice solving problems and prepare for the corresponding quiz and the final exam. As mentioned in the studies, the problem sets are not graded or collected.

Each unit you will have an opportunity to discuss the problem sets with your fellow learners. Use this discussion to ask questions, make comments, ask for help, and assist your peers with the problem sets. Teamwork is encouraged.

Four posts are **required** in this discussion: two initial and two response. Additional posts are optional and recommended, but not required for your grade:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the assigned problem set that you are confused about, write it out fully and show any work you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are response posts. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u01a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 1.1, page 12, problems 15, 53.
- Section 1.2, page 19, problems 21, 30.
- Section 1.3, page 30, problem 72 only.
- Section 1.4, pages 35–36, problems 21, 28.
- Section 1.5, pages 48–49, problems 36, 74.
- Section 1.6, page 55, problems 24, 28.
- Section 11.1, page 537, problem 11.
- Section 11.2, pages 548–549, problem 2.

u01a2 - Boolean Circuits and Technology

This is the first of five independent application assignments. Each assignment will allow you to learn how the topics of this course apply to the areas of computer science, Internet technology, or technology applications.

For this assignment, imagine that we are many years into the future and you have been hired by a technology company to create a human door key system. When a person steps on a special mat containing sensors that is located at his or her front door, these sensors grab information, send it through a set of circuits, and reach a one of two logical conclusions:

- Yes (or 1), the person lives in this house and may enter.
- No (or 0), this person does not live in this house and may not enter.

If the result is yes, the door automatically opens. If the result is no, you can decide what happens.

For this assignment, complete the following:

1. **Define at least five features** that can be sensed by the magic door mat such that each feature has a result of 0 or 1. This magical mat can sense anything, such as weight, height, eye color, hair color, gender, and so on. Because the result of the sensor can only be 0 or 1, you will have to think about how to do this. As an example, if the weight is greater than 125, the sensor returns a 1, else a 0. Similarly, if the hair color is NOT brown, the sensor returns a 0, else 1. These are just some ideas.
2. Explain each of your features clearly, including exactly what the 0 or 1 represents in each case.
3. **Using *at least five gates and* at least five inputs,** create a logical circuit that describes your system. This circuit will have at least five inputs, at least five gates, and one—and only one—output. Remember, the input values to and from the gates are only 0 or 1, but you can name them according to what they represent.
4. Create a logic table to show some examples of people who will and will not be able to enter. This table will be labeled with your attributes and will contain values of only 0 and 1.
5. Describe your circuit using a Boolean expression with proper use of AND, OR, and NOT symbols.
6. Describe what will happen to a person who cannot enter (that is, when the output of the circuit is 0). This is up to you, so feel free to be creative. Include at least one paragraph explaining your circuit, your features, and what will happen in cases where a person can or cannot enter the door. *Hint: You can have the door inform the person that they cannot enter and why.*

You may find it useful to read Chapter 11, pages 532–542, and look at the examples. You can combine attributes to make sure only the right person gets in. For example, if a person's weight is greater than or equal to 125, you can have a 1. If the weight is less than 125, you can have a 0. Therefore, if the person weighs 250 pounds, this would result in a 1. You can use the circuit creatively with AND, OR, and NOT gates to grant or deny access using a set of measurable attributes.

Review the Boolean Circuits and Technology Scoring Guide to understand how you will be graded on this assignment.

Unit 2 >> Relations, Functions, and Relational Databases

Introduction

This unit introduces the concepts of functions, relations, sequences, strings, and the application of relational databases.

In Unit 1, you learned about sets, or collections of objects. You will see in Chapter 3, which is assigned in this unit, that relationships can be defined between sets of values or objects. These sets can be finite or infinite, and the relations defined can be sequences, functions, or mappings.

As an example, suppose you are given a function *f* on a value *x*, *f*(*x*). Next, suppose the *f*(*x*) is defined as:

$f(x) = 3x + 4$.

Here, the function *f* is an infinite set of real numbers (the range) that is mapped to or defined by an infinite set of real numbers (the domain) such that for any value *x* in the domain, the value of *f*(*x*) is mapped to or defined by 3 *x* + 4. This function tells us that the value 1 is mapped to the value 3(1) + 4, or 7. We can also say that *f*(1) = 7. Relationships between objects or values can be defined in many ways, and you will look at examples of functions, sequences, modulus, hashes, strings, and database applications.

Functions

A function is a special type of relation. A function assigns or maps each value in the domain (usually known as *x*) to exactly one member of the range (usually known as *y*). Functions can be finite or infinite.

$f(x) = x^2$ is an infinite function.

Sequences

A sequence is a special type of function, which makes it a relation. For a sequence, the domain is always a set of consecutive integers, such as {1, 2, 3, 4}. The range is a function of that set of integers and is often denoted as C_n, where n is the nth term in the sequence.

Relations and Equivalence Relations

A relation is also a mapping from one set to another, but with fewer constraints than those of functions and sequences. There are different types of relations:

- Binary: A relationship or mapping between two items.
- n-ary: A mapping or relationship between multiple items, objects, or numbers.
- Equivalence: A relation that is reflexive, symmetric, and transitive.

Applications: Matrices and Databases

A matrix is an organized method for storing values and processing these values. A matrix can be used to represent a tree or graph, and it can be used to plan or evaluate an algorithm.
A database is a structure for storing data, as well as accessing and processing that data. Relations are directly related to databases as most databases are constructed using relationships between entities and items. These are known as relational databases.

Learning Activities

u02s1 - Studies

Introduction

Remember to read the introduction prior to starting the work in this unit.

Multimedia

Click **Functions and Relations** to view the presentation, which introduces this unit's topics, including:

- Functions.
- Modulus operator.
- One to one functions.
- Onto functions.
- Composite functions.
- Sequences.

- Sigma notation.
- Relations.
- Order relations.

Reading

Read the following in your *Discrete Mathematics* text:

- Chapter 3, "Functions, Sequences, and Relations," pages 111–172. In addition to focusing on functions, sequences, and relations, you will learn about equivalence relations, matrices of relations, and relational databases.

u02s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on functions, strings, relations, and relational databases. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 3.1, pages 125–126, exercises 19 and 64.
- Section 3.2, pages 139–140, exercises 128 and 147.
- Section 3.3, page 150, problem 36.
- Section 3.4, page 156, problems 3 and 8.
- Section 3.5, page 164, problems 2 and 9.
- Section 3.6, page 169, problem 13.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in this unit, which covers all topics in Units 1 and 2, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 2** to learn a step-by-step process for solving each problem for this set.

u02d1 - Relational Databases

Relations and functions play important roles in computer applications, systems management, database management, and error management. In this discussion, you will look at relational databases and their functions and applications.

Databases are common computer and Internet technology applications. Online banking, shopping, iPhones, music, video, personal files, businesses, governments, and academic institutions all use databases to store, process, and retrieve information, including images, video, and music.

Suppose you are a database manager and oversee a large database with the following four tables: EMPLOYEE, DEPARTMENT, SUPPLIER, and BUYER.

Now, turn to page 168 in the text. Under Exercises, you will see these four tables listed as 3.6.4, 3.6.5, 3.6.6, and 3.6.7.

Part 1: Table and Database Operations

Complete the odd problems from 5 through 15 on page 169.

1. Write out each question fully.
2. Write a sequence of table operations to answer the query required for each question.
3. Provide the final answer to the query for each question.

Part 2: Researching the Web for Database Applications

1. Create two paragraphs that describe two current database application packages. For example, you might choose Oracle and SQL Server.
2. Read about what these packages offer, how they are alike, how they are different, and what applications they are best suited for.
3. What language is commonly used to manipulate and communicate with database applications? Offer a few examples of statements from this language.

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Do you agree with your peer's assessment of the problem? Explain.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u02d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post,

write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u02a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 3.1, pages 125–126, problems 21, 63.
- Section 3.2, pages 137–140, problems 27, 148.
- Section 3.3, page 150, problem 31.
- Section 3.4, page 156, problems 2, 12.
- Section 3.5, pages 163–164, problems 3, 10.
- Section 3.6, page 169, problem 12.

u02q1 - Quiz 1

You have been introduced to various discrete math methodologies, and this multiple-choice quiz will gauge your understanding of the material presented in the first two units of this course. The quiz provides an opportunity for you to demonstrate your mastery of the following course competencies:

- Describe how discrete math theories are used in computer science.

- Apply the methodologies of discrete math.
- Use mathematical reasoning, deductive and inductive logic, and proofs to solve problems.

Read the following instructions and parameters before taking the quiz:

- There is no time limit on the quiz.
- Once you have answered each question, submit your quiz to receive credit. It is automatically scored, so you will receive feedback immediately. There are 100 points possible. Each question is worth 10 points.
- You may access the quiz to review the questions after you have taken it; however, you cannot retake the quiz to change your grade. Once the grade is recorded, it cannot be changed.
- Since you can take the quiz only one time and during a single sitting, do not begin the quiz until you are ready to complete it. Do not forget to submit the quiz once you complete it.
- When you access the quiz, read the Instructions and Troubleshooting page prior to starting the quiz.
- Remember, the quiz will focus on topics directly covered in the readings and problem sets for Units 1 and 2.
- If you have questions or concerns, contact your instructor immediately.

Click the linked quiz title to access the quiz. If you have any issues with the quiz, contact your instructor.

Unit 3 >> Counting, Combinations, and Permutations

Introduction

This unit focuses on probability theory, discrete probability theory, counting rules, permutations, and combinations. First, let us think about basic discrete (noncontinuous) probability using a standard deck of playing cards. We know there are 52 cards, 13 different values from 2 to ace, and four suits (that is, clubs, spades, hearts, and diamonds). Many probability questions can be asked about a deck of cards. For example, what is the probability that a card you choose is a queen?

P(card = queen).

We know there are 52 cards and four queens.

P(card = queen) = 4/52.

We can extend this concept to multiple events. What is the probability of choosing two cards and both are queens?

P(card1 = queen) AND *P*(card2 = queen).

Notice the AND. You will see on page 266 of your course text that there is a concept called the multiplication principle that considers the results of more than one event occurring together.

P(card1 = queen) AND *P*(card2 = queen) = 4/52 * 3/51 = 12/2,652.

We used multiplication in this case and did not replace the first card after choosing it.

Similarly, the addition principle is used for independent events such as the probability of choosing a card and it being a queen OR a king.

P(card1 = queen) OR *P*(card1 = king) = 4/52 + 4/52 = 8/52.

Set Theory and Probability Theory

In Chapter 6, you will find that set theory and probability theory are interrelated. An event can be thought of as a set of possible outcomes. For example, rolling one six-sided die is an event. This event is discrete because each outcome is a whole number (that is, no fractions or decimals).

The event *E* = {1, 2, 3, 4, 5, 6}.

Rolling a die is an event that offers a set of six possible outcomes as noted above. This set can be named *E*. This set *E* follows all rules of set theory, such as NOT *E* being all values not in the set *E* given a universe *U* of values.

Not all probability is discrete. If an event *E* can have any one of a set of continuous values, then that type of probability falls under the category of continuous probability and is generally taught in a class about statistics and probability. As an example, suppose you are measuring the speed of cars passing a certain point on a highway. The event is the speed of the car being measured. This measurement can be any real value between zero and 120 miles per hour (mph), including decimal values. Therefore, a car can be driving 57.9 mph. This is different from discrete probability because, for example, rolling a die can only result in a whole number. There is no way to roll a 3.6. In this class, we will focus on discrete probability.

Permutations and Combinations

A permutation is the number of possible ways of rearranging a discrete set (no decimals or fractions) of objects. For example, if I gave you four pictures to hang on the wall and asked how many ways you could hang them in a straight line, you would tell me:

$P(4, 4) = 4! = 4 * 3 * 2 * 1 = 24$ ways.

The $P(4, 4)$ notation reads, "4 items, permute all 4 of them." The 4! reads, "4 factorial" and is a mathematical construct noted on page 271, Theorem 6.2.10. The general notation of permutation is $P(n, r)$, given n objects, how many ways can you permute r of them.

A combination is the number of ways you can select a set of objects when the order does NOT matter. The notation is $C(n, r)$ and formula is on page 273, Theorem 6.2.16.

Computer Applications

The areas of computer science and information technology (IT) often use probability theory. For example, the popular Google Maps and related iPhone applications allow users to map and get directions (shortest path) to just about any location. The study of combinations allows for the calculation of the number of paths or routes between any two points given a map or grid.

Discrete Probability Theory

In sections 6.5 and 6.6 of your text, pages 297 through 313, you will study rules and theories related to probability. Note that an event is the outcome or combination of outcomes of an experiment. All outcomes make a set known as the sample space. For any outcome x in a given sample space, a function P, known as the probability function, assigns a value to $P(x)$.

Note that: $0 \leq P(x) \leq 1$.

This tells us that the probability of any event must be between 0 and 1 (or 0% chance to 100% chance).

For example, let S be a sample space with all possible values of rolling a six-sided die:

- $S = \{1, 2, 3, 4, 5, 6\}$.
- Let $x = 4$.
- $P(4) = 1/6$, which is between 0 and 1.

Learning Activities

u03s1 - Studies

Introduction

Review the introduction prior to starting the work in this unit.

Multimedia

Click **The Counting Principle** to view the presentation, which introduces you to this unit's topics, including:

- Multiplication and addition principles.
- Permutations and combinations.
- Catalan numbers.
- Discrete probability.
- Conditional probability.
- Pigeonhole principle.

Readings

Read the following from your *Discrete Mathematics* text:

- Chapter 6, "Counting Methods and the Pigeonhole Principle," sections 6.1, 6.2, 6.3, 6.5, and 6.6, pages 255–313. Topics in these sections include permutations, combinations, and discrete probabilities. You will also investigate the discrete probability theory.

u03s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on counting, combinations, and permutations. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 6.1, pages 264–265, exercises 8 and 39.
- Section 6.2, page 278, exercises 14 and 41.
- Section 6.3, page 288, exercise 8.
- Section 6.5, page 299, exercises 3, 7, 19, and 22.
- Section 6.6, pages 311–312, exercises 3, 16, 30, and 61.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in Unit 4, which covers all topics in Units 3 and 4, and for the cumulative final exam in Unit

10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 3** to learn a step-by-step process for solving each problem for this set.

u03d1 - Networking and Probability

Probability and statistics allow us to make decisions when dealing with uncertain variables. For example, assume you are the network administrator of a large firm and have ordered 100 RAID (redundant array of independent disks) devices. You would like to make sure all of the RAIDs work before sending them to different departments. But you do not have time to check all of the devices. If you check some of them and show they work, how confident could you be that all 100 devices work?

Of course, testing all of the RAIDs would be the best way to be confident, but assume you do not have the time. If you only have time to check 5 of the 100, and none of the 5 are defective, how confident would you be that none of the other microprocessors are defective (that is, all 100 devices are fully functional)?

Answer the following questions, which will help you understand the problem. Explain each question and your answer:

1. How can you use discrete probability to calculate the probability of testing 5 devices such that all 5 end up being nondefective RAIDs, given there are 20 defective RAIDs total of the original 100 RAID devices?
2. What is the sample space in this problem?
3. Is this a ratio of combinations or permutations problem? Explain why and show an example. *Hint: Read section 6.5 and see Example 6.5.4 in your textbook.*
4. Assume there are exactly 5 defective RAIDs out of the 100 RAID devices. What is the probability of testing 20 RAIDs at random and finding all of them nondefective?

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Do you agree with your peer's assessment? Explain.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u03d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this

area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u03a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 6.1, pages 264–265, problems 2, 38.
- Section 6.2, page 278, problems 12, 40.
- Section 6.3, page 288, problem 2.
- Section 6.5, page 299, problems 2, 6, 16, 21.
- Section 6.6, pages 311–312, problems 2, 15, 29, 58.

u03a2 - Web Addresses and Discrete Math Applications

The World Wide Web is a boundless resource of information. It contains blogs, tutorials, demos, videos, text, references, and resources for just about any topic you may want to research. Computer users visit Web sites identified by an address, or URL (uniform resource locator). You are doubtless familiar with URLs, such as http://www.cnn.com or http://www.fox.com.

Unlike URLs, which use letters, the DNS (domain name system) resolves URLs into a numerical IP (Internet provider) address. These addresses are assigned to nodes in a computer network. While

you answer the following questions, ask yourself if there are an infinite number of Web sites available.

Define the following:

- What is a URL?
- What is an IP address?
- What is a DNS?

Explain:

- How are URLs assigned to IP addresses?
- What does DNS stand for and what does it do?
- Is there a limited number of Web sites available using IP version 4 (documented as IPv4)?
- What changes have been made in IPv6 compared with IPv4?

Calculate:

- Given that an IP address in IPv4 is a 32-bit string, how many different addresses can be encoded? Show your calculation. *Hint: Use the multiplication principle.*
- How many different Web sites could be encoded using IPv6? Show your calculation.

For this assignment, make sure you clearly and completely define, explain, and show all work for your calculations. The assignment should be two to three pages. Include all resources used. Review the Web Addresses and Discrete Math Applications Scoring Guide to understand how the assignment will be graded.

Unit 4 >> Proofs and Reasoning, Induction, and Deduction

Introduction

In Unit 4, you will look at proofs, mathematical systems, and counter examples. In previous chapters, you examined the use of logic to reach conclusions. Creating a proof required a series of logical steps that together lead directly to proving the result.

A mathematical system consists of axioms, definitions, and undefined terms. Axioms are always assumed to be true. Undefined terms are generally implicitly defined by axioms, and definitions are

used to create new concepts. A theorem is a proposition that has been proved to be true, types include lemmas and corollaries. Examples 2.1.1 and 2.1.2, on page 63, offer examples of axioms and a mathematical system using the real numbers, respectively.

Direct Proofs

A direct proof assumes a set of axioms and uses logic and reasoning, with a clear sequence of steps to offer a unambiguous proof of a claim or statement.

> Example: Give a direct proof that for all integers m and n, if m is odd and n is even, then $m + n$ is odd.
>
> Given: m is odd and n is even.

- Step 1: Suppose we have an integer k such that $m = 2k + 1$. Note that, in this case, no matter what the value of k, m will always be odd as required by the given terms of the statement.
- Step 2: Suppose we have an integer w such that $n = 2w$. Here again, no matter what the value of w, n will always be even as we want it to be.

We have defined an odd m and an even n.

Our next step is to add them together as this is what we are trying to prove.

$$m + n = (2k + 1) + 2w = 2k + 2w + 1 = 2(k + w) + 1.$$

Why did we do this? Because we know that if we can write $m + n$ as two times something plus 1 that it must be odd. For the full proof, please see page 69.

Proof by Contradiction and Contrapositive

Proof by contradiction and contrapositive is a method that starts by assuming that a statement is not true, but then shows that it is in fact true because a known axiom will be false otherwise. Example 2.2.1 on page 72 is a great example of this concept. Similarly, proof by contrapositive is a special case of proof by contradiction. As you will see in sections 2.2 and 2.3, pages 72–88, there are many different proof methods, each with its own unique characteristics.

Proof by Induction

Proof by induction is a type of proof method that uses a few specific steps, such as the basic step, the inductive step, and finally the conclusion. The basic step is the base case. For example, use induction to prove that $5n - 1$ is divisible by 4 for all $n \geq 1$.

The base case is to show that the statement is true for a small and easy-to-prove value, such as 1.

When $n = 1$.

$5^n - 1 = 5^1 - 1$, which is 4, and we know that 4 is divisible by 4, so our base case is true for $n = 1$.

The next step is to assume that our claim is true for some value of n, call it m.

So, we assume that:

$5m - 1$ is divisible by 4 for some value $m > 1$.

Now, our inductive step uses our assumption and our base step and proves that our statement is true for all values larger than m, say $m + 1$:

$5(m + 1) - 1$.

The key is to turn the above statement with $m + 1$ into a combination of statements that we know are true:

$5(m + 1) - 1$ can be written as $5m * 5 - 1$, which can be written as $4 * 5m + 1 * 5m - 1$, which is the same as:

$4 * 5m + 5m - 1$.

We know that $4 * 5m$ is divisible by 4 because it is a factor of 4. We also know from our assumption that $5m - 1$ is divisible by 4 because this was our assumption. Finally, we know that the sum of two values divisible by 4 must itself be divisible by 4. For the full proof, please see pages 98 and 99.

Proofs are never easy and require practice, patience, and experience!

Learning Activities

u04s1 - Studies

Introduction

Remember to read the introduction before completing the unit activities; it will give you a foundation for the work you will be doing in this unit.

Multimedia

Click **Proofs** to view the presentation, which introduces this unit's topics, including:

- Definitions.
- Types of proofs.
- Rules of influence.
- Rules of inference for quantified statements.
- Resolution proofs.
- Mathematical induction.

Reading

Read the following in your *Discrete Mathematics* text:

- Chapter 2, "Proofs," sections 2.1 through 2.4, pages 63–102. In these sections, you will focus on mathematical systems, counterexamples, direct proofs, resolution proofs, and various methods of proofs.

u04s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on proofs, reasoning, induction, and deduction. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 2.1, page 70, exercises 6 and 12.
- Section 2.2, page 81, exercises 8 and 12.
- Section 2.3, page 88, exercise 4.
- Section 2.4, page 96, exercises 5 and 8.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in this unit, which covers all topics in Units 3 and 4, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 4** to learn a step-by-step process for solving each problem for this set.

u04d1 - Machines, Intelligence, and Neural Networks

In this discussion, you will research the history of artificial intelligence (AI). Write two to three paragraphs on the history of AI and how the ideas of logic, resolution, and unification were implemented within machines so they could demonstrate "intelligence." In your post, answer at least two of the following questions:

1. What is the resolution and unification algorithm, and what is an algorithm?
2. What is the Turing test, and who is Alan Turing?
3. What is a neural network?
4. Can machines really demonstrate intelligence?

In order to answer these questions, you will have to search the Internet. Suggested search terms include history of artificial intelligence, Turing test, and so on.

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Assess your peer's post. Do you agree or disagree? Explain.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u04d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u04a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 2.1, page 70, problems 5, 11.
- Section 2.2, page 81, problems 9, 14.
- Section 2.3, page 88, problem 6.
- Section 2.4, page 96, problems 3, 9.

u04q1 - Quiz 2

This multiple-choice quiz will gauge your understanding of the material presented in Units 3 and 4. The quiz provides an opportunity for you to demonstrate your mastery of the following course competencies:

- Apply the methodologies of discrete math.
- Use mathematical reasoning, deductive and inductive logic, and proofs to solve problems.
- Apply discrete math methods and tools to solve problems encountered in a work setting.

Read the following instructions and parameters before taking the quiz:

- There is no time limit on the quiz.
- Once you have answered each question, submit your quiz to receive credit. It is automatically scored, so you will receive feedback immediately. There are 100 points possible. Each question is worth 10 points.
- You may access the quiz to review the questions after you have taken it; however, you cannot retake the quiz to change your grade. Once the grade is recorded, it cannot be changed.
- Since you can take the quiz only one time and during a single sitting, do not begin the quiz until you are ready to complete it. Do not forget to submit the quiz once you complete it.
- When you access the quiz, read the Instructions and Troubleshooting page prior to starting the quiz.
- Remember, the quiz will focus on topics directly covered in the readings and problem sets for Units 3 and 4.
- If you have questions or concerns, contact your instructor immediately.

Click the linked quiz title to access the quiz. If you have any issues with the quiz, contact your instructor.

Unit 5 >> Pseudocode, Algorithms, and Recursion

Introduction

In this unit, we will talk about algorithms, including recursion, as well as pseudocode, control statements, and basic programming (that is, coding) principles. This unit is particularly important to the areas of IT and computer science as both use a number of different programming languages to complete tasks via algorithms. Here, we will focus on understanding how to interpret and create algorithms, how to read and understand pseudocode (that is, English-like programming code), and how to write recursive and regular pseudocode. In Unit 6, we will use this knowledge and learn how to analyze algorithms in time and space.

Algorithms

An algorithm is a method that will successfully complete a specific task. The key difference between an algorithm and a heuristic is that an algorithm will guarantee a solution, even if it takes 1 million years. A heuristic might be a lot faster, but might not always produce a solution, or the optimal solution. In this unit, we will focus only on algorithms. Specifically, an algorithm is a step-by-step process or method for completing a task, solving a problem, or determining a solution. An algorithm can have one or more inputs as well as one or more outputs.

A simple example might be to program a computer to add two numbers and output the sum. The input would be the two numbers; the output would be the sum. The algorithm would be the entire method and code that accepted the input and produced the expected correct output. In Unit 6, we will learn that, while there are often many methods for solving the same problem, some methods are faster and require less space. This is very important in IT as space and time are always limited.

Pseudocode

There are many different programming languages, each with specific strengths and attributes. Examples are Java, C#, Perl, Python, SQL, and Visual Basic. In this course, we will use an English-

based language called pseudocode. The idea is that we can use pseudocode to create algorithms without having to worry about the specific syntax and the memory usage of a programming language or related environment.

Algorithm 4.1.1, on page 174, is a great example of the use of pseudocode to show the steps to find the maximum of three numbers. The pseudocode shows the input, the steps of the algorithm, and the output. When writing pseudocode, it is best to show every relevant step so that your pseudocode could easily be converted to an actual programming language if desired.

Appendix C

Appendix C starts on page 620 and covers a lot of information about pseudocode, including the if, then, else construct; the while loop construct; and the for loop construct. These are highly important to understand. For example, all programming languages have a form of the if construct.

> if (condition) then (action) else (action).

The else is not required, but it is optional as needed. Be sure you are familiar with Appendix C.

Recursive Algorithms

There are many types of algorithms. One is recursive, and it is unique as it uses the concept of calling itself. Here is an example.

We know from earlier units that a factorial (!) is the decreasing product of the given number, such that:

> 7! = 7 * 6 * 5 * 4 * 3 * 2 * 1.

We can write a program (create pseudocode in our case) to take as input any number *x* and offer as output the value of the factorial *x*! Pseudocode for this example can be found on page 214. The key is that we name a function that performs a task:

> *factorial*(*n*) {if(*n* == 0) return 1 else return *n* * *factorial*(*n* – 1)}.

Notice that inside the function called *factorial*(*n*), we make a call to that same exact function unless the value of *n* is 0. Also notice that when we call the function *factorial*(*n* – 1), we also reduce the value of *n* by 1.

The best way to really understand an algorithm is to step through it by hand. Let us do that here:

> Suppose we start with *n* = 7.

This means that the value of 7 is passed to the function factorial as input. The first statement asks if the value of *n* is equal to 0. It is not because it is 7, so we move to the next step. The next step tells us

to multiply the value of *n* (at this time 7) to the value that we get when we call the factorial function using the value *n* – 1, or 6.

So, the program, in memory, saves the value 7 and then calls the program with the value of 6. We repeat the steps. Because 6 does not equal 0, we move to the next step that says multiply 6 by the value you get when you call the function factorial on 6 – 1, or 5. This continues until *n* is decreased to 0 and the program ends.

The recursive portion has been storing in memory each value, 7 * 6 * 5 * 4 * 3 * 2 * 1 and then, once *n* = 0, it returns the product.

Recursion is not a simple concept and takes some getting used to. Review the rest of the examples in section 4.4, pages 204–212.

Time Complexity

Because computers are fast, we generally use upper bounds to measure the speed of a program. For example, we might say that a specific sorting algorithm runs in $O(n^2)$ time or perhaps even faster at $O(n \log n)$ time. The big *O*, or *O*( ), is a notation that notes the upper bound, or the closest fit. When we say a program runs in $O(n^2)$ time, we mean that for an input of size *n*, we can expect the algorithm to take approximately, or on the order of, n^2 time to run. Generally, the unit of time referred to it the microsecond. However, keep in mind that this concept can be used for any unit of time and that this unit of time is largely determined by the clock speed of the computer processor. As we know, some computers are faster than others. The big *O* is actually known as an asymptotic upper bound.

The table below, also shown on page 184, Table 4.3.1, shows a number of run speeds and the relationships between them, such that $O(n)$ is faster than $O(n \log n)$ is faster than $O(n^2)$, and so on. Example 4.3.1, on page 185, shows how a bound is assigned. Figure 4.3.1, on page 188, is an excellent graph of the relationship between run times and speeds. It is imperative to always complete and review the text examples.

Learning Activities

u05s1 - Studies

Introduction

Be sure to read the unit introduction, which addresses algorithms, pseudocode, and recursive algorithms.

Multimedia

Click **Algorithms** to view the presentation, which introduces this unit's topics, including:

- Characteristics of algorithms.
- Notation for algorithms.
- The Euclidean algorithm.
- Recursive algorithms.
- Complexity of algorithms.

Readings

Read the following in your *Discrete Mathematics* text:

- Chapter 4, "Algorithms," pages 173–211, sections 4.1 through 4.3. In this chapter, you will look at examples of algorithms, analyze algorithms, and work with recursive algorithms.
- Appendix C, "Pseudocode," pages 620–624. In this section, you will learn about the pseudocode that is used in the text.

u05s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on pseudocode, algorithms, time complexity, and recursion. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 4.1, page 176, exercise 7.
- Section 4.2, page 183, exercises 2 and 10.
- Section 4.3, page 198, exercises 3, 12, 18, and 22.
- Appendix C, page 625, exercises 3 and 9 (make sure you complete understand how these problems work).

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in Unit 6, which covers all topics in Units 5 and 6, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem

sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 5** to learn a step-by-step process for solving each problem for this set.

u05d1 - Data Mining, Time Complexity, and Algorithms

In Appendix C, you read about different programming control structures used to write pseudocode and actual computer algorithms, such as if statements, while and for loops, and function calls. For this discussion, assume you work for a data mining company and your job is to write a program to find information on various Web sites pertaining to sales of the Lenovo X200. After your algorithm finds this data, more complex analysis will be done to extract more meaningful information from the data.

Your algorithm is going to scan different sites and search for the character string "Lenovo X200." Assume you decide to use an algorithm similar to Text Search (see algorithm 4.2.1 on page 178 of your text for an explanation of what this is). If the algorithm finds a site that contains the string (that is, Lenovo X200), assume that it then stores all data or all the text on that particular site into a storage area.

To understand this problem fully, answer the following questions:

1. What is data mining?
2. What is a character string?
3. What is the worst case run time of this algorithm in terms of *p, m, t, n* (that is, what is *O*)?
4. How long do you think it will take this algorithm to run? Note the time complexity as *O* (run time in terms of *n*).
5. Assume that each Web site, on average, has character strings of length 10,000 and that the length of the character string "Lenovo X200" is 11. How many computations will the algorithm need to make per site?
6. Why is speed and the analysis of algorithm speed so important?

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Did you arrive at the same time complexity calculation? Explain why or why not.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u05d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post,

write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u05a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 4.1, page 176, problem 5.
- Section 4.2, page 183, problems 5, 9.
- Section 4.3, page 198, problems 2, 15, 19, 21.
- Appendix C, page 625, problems 2, 8.

u05a2 - Algorithms and Time Complexity

In algorithm development, the time and space required for algorithm completion is paramount. As users, we know that when a computer process takes too long, we try to avoid it. This truth encourages all IT and computer-based companies to produce faster products and services.

For this assignment, write a one- to two-page paper that includes all required algorithms and pseudocode describing the time and space complexity of algorithms. Include the following:

1. Answer the following questions:

- What is time complexity?
- What is space complexity?

2. Compare and contrast polynomial time algorithms and nondeterministic polynomial (NP) time algorithms (one paragraph minimum).
3. Provide an example of an algorithm for each worst-case run times:
    - $O(n)$.
    - $O(n^k)$. *Note that this is called polynomial-time, where* k *is any number greater than 1.*
    - NP-time.

Hint: Quick sort is an algorithm that runs in $O(n \log n)$ time.

Review the Algorithms and Time Complexity Scoring Guide to understand how the assignment will be graded.

Unit 6 >> Time Complexity, Recurrence Relations, and Analysis of Algorithms

Introduction

In this unit, we will continue our discussion of algorithms, extending and turning our focus to analyzing, measuring, and comparing them. We will review section 4.3, "Analysis of Algorithms," and then move on to Chapter 7, which addresses recurrence relations. We will also examine cryptography, a popular computer science application. We will address this application in a unit discussion.

Recurrence Relations

Sometimes, determining the run time of an algorithm, especially a recursive algorithm, might take further calculations. As we have seen, some programs can be quite involved and even call themselves. Recurrence relations can be used to analyze more complicated algorithms. They can also be used to investigate the relationship between input values in a sequence.

For example, suppose we have a simple algorithm that starts with 5, then adds 3 to get the next term, and repeats on the next term. This algorithm would have an input of 5 and the following output:

5, 8, 11, 14, 17, and so on.

Another way to write this algorithm would be to denote terms with a $_n$ values:

$a_1 = 5.$

$a_2 = 8.$

$a_3 = 11.$

Given this notation, how can we determine a_{101}?

This is where recurrence relations come into play. We know the base case: $a_1 = 5$. We also know the recurrence relation:

$a_n = a_{n-1} + 3.$

Let us start to write these out to see if we can come up with an equation:

$a_n = a_{n-1} + 3.$

$a_n = a_{n-2} + 3 + 3.$

$a_n = a_{n-3} + 3 + 3 + 3.$

$a_n = a_{n-4} + 3 + 3 + 3 + 3$ and so on until a_{n-x} is 5 and we stop.

First, notice that we are expanding this recursive (recurrence) equation. Placing parentheses provides us with a better look:

$a_n = (a_{n-1}) + 3.$

$a_n = ([a_{n-2}] + 3) + 3.$

$a_n = ([\{ a_{n-3}\} + 3] + 3) + 3.$

$a_n = ([\{ a_{n-4} + 3\} + 3] + 3) + 3$ and so on until a_{n-x} is 5 and we stop.

If we continue, we would next expand the a_{n-4} term into $a_{n-5} + 3$ and so on. This will continue until we reach the last term. However, happily, we do not have to continue. We expand only until we see the following pattern:

$a_n = a_{n-x} + (3 * x).$

The x represents the number of threes that we have. Notice that for $a_n = ([\{ a_{n-4} + 3\} + 3] + 3) + 3$, we have four threes and an a_{n-4} term. So, we use x to represent this concept, as we know we get one

more threes each time we expand and our a_{n-4} term will become a_{n-5}.

$$a_{101} = a_{101-100} + (3 * 100) = a_1 + 300 = 5 + 300 = 305.$$

Recall that a_1 was defined as 5.

So, to review, a recurrence equation can be used to find the value of any item in the sequence. However, to do this, the recurrence equation must be solved and evaluated. This is the topic of section 7.2, starting on page 338.

Recurrence equations can be used to determine how long an algorithm will take to run. Algorithm 7.3.2, on page 354, looks at the binary search. Example 7.3.3, on page 354, analyzes this search method.

Cryptography

The word *cryptography* comes from the word *cryptic*, which means hidden. Cryptography focuses on encrypted and decrypted information that is intended only for the receiver. E-mail systems, such as Hotmail, claim to use encryption. Cryptography uses many areas of discrete math, including modulus, prime factorization, and greatest common divisors. Chapter 5 has information about this applications area.

Learning Activities

u06s1 - Studies

Introduction

Read the unit introduction prior to starting any activities. It includes a discussion of recurrence relations.

Multimedia

Click **Number Theory and Recurrence Relations** to view the presentation, which introduces this unit's topics, including:

- Number systems.
- Hexadecimal number system.
- Recurrence relation.
- Fibonacci sequence.

- Towers of Hanoi.
- Solving recurrence relations.

Readings

Read the following in your *Discrete Mathematics* textbook:

- Chapter 4, "Algorithms," pages 173–212. You will find reviewing section 4.3 useful in completing your studies and work. Section 4.4 focuses on Recursive Algorithms.
- Chapter 5, "Introduction to Number Theory," pages 214–254. In this chapter, you will focus on divisors, representations of integers and integer algorithms, the Euclidean algorithm, and the RSA public-key cryptosystem.
- Chapter 7, "Recurrence Relations," pages 327–372. The chapters focus on solving recurrence relations and applications to the analysis of algorithms.
- Read "Binary and Hexadecimal Expansions and Operations" to learn about binary, decimal, and hexadecimal number systems and Microsoft Excel's conversion tool.

u06s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on time complexity, recurrence relations, analysis of algorithms, and cryptography. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 4.4, page 210, exercises 5 and 11.
- Section 7.1, page 335, exercises 3, 11, and 24.
- Section 7.2, page 347, exercises 6 and 13.
- Section 7.3, pages 360–363, exercises 2 and 45.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in this unit, which covers all topics in Units 5 and 6, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 6** to learn a step-by-step process for solving each problem for this set.

u06d1 - Cryptography and Security

Cryptography is an exciting IT field that focuses on secure communications, typically on a network. Many encryption methods use number theory to encode and decode messages. This number theory typically draws upon ideas such as prime factorization, greatest common divisors (GCD), modulus, and so on.

For this discussion, read Chapter 5, section 5.4, which starts on page 250 of your text, which discusses number theory and applications to cryptography. Post an explanation of how the RSA public-key cryptosystem works that includes answers to the following questions:

1. What is a key?
2. What are public and private keys, and how are they used in the RSA public-key cryptosystem?
3. How are messages encoded and decoded using mod and GCD?
4. If you are using $p = 17$, $q = 23$, and $n = 31$, what would the encryption of 101 be, using public key $z = pq$?
5. Where is security used, and why is it important to everyone?

Review the Discussion Participation Scoring Guide prior to posting. Make sure you include a minimum of three posts and that your first, or main, post responds to all parts of the question posed.

Response Guidelines

Read your peers' posts and respond to two. Did you reach the same encryption as your peer? Explain why or why not. Show your work.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u06d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u06a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 4.4, page 210, problems 6, 12.
- Section 7.1, page 335, problems 2, 6, 20.
- Section 7.2, page 347, problems 3, 6.
- Section 7.3, pages 360–362, problems 3, 43.

u06q1 - Quiz 3

This quiz will gauge your understanding of the material presented in Units 5 and 6. The quiz provides an opportunity for you to demonstrate your mastery of the following course competencies:

- Describe how discrete math theories are used in computer science.
- Apply methodologies of discrete math.
- Use mathematical reasoning, deductive and inductive logic, and proofs to solve problems.
- Apply discrete math methods and tools to solve problems encountered in a work setting.

Read the following instructions and parameters before taking the quiz:

- There is no time limit on the quiz.
- Once you have answered each question, submit your quiz to receive credit. It is automatically scored, so you will receive feedback immediately. There are 100 points possible. Each question is worth 10 points.
- You may access the quiz to review the questions after you have taken it; however, you cannot retake the quiz to change your grade. Once the grade is recorded, it cannot be changed.
- Since you can take the quiz only one time and during a single sitting, do not begin the quiz until you are ready to complete it. Do not forget to submit the quiz once you complete it.

- When you access the quiz, read the Instructions and Troubleshooting page prior to starting the quiz.
- Remember, the quiz will focus on topics directly covered in the readings and problem sets for Units 5 and 6.
- If you have questions or concerns, contact your instructor immediately.

Click the linked quiz title to access the quiz. If you have any issues with the quiz, contact your instructor.

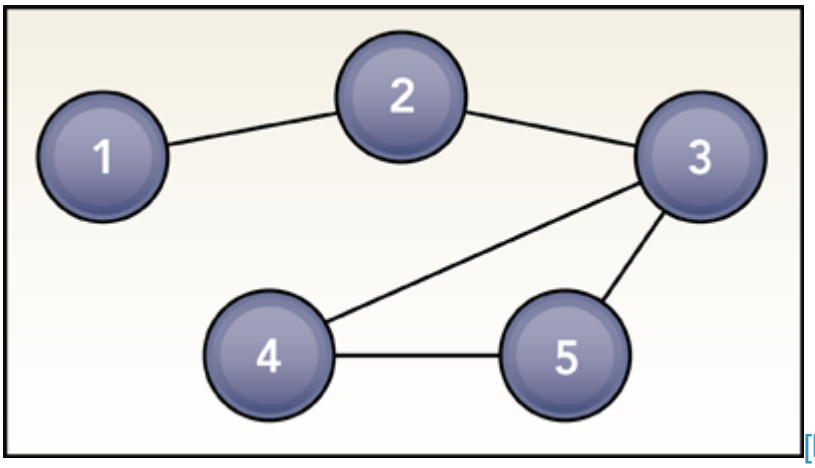
Unit 7 >> Graphs and Adjacency Matrices

Introduction

In Unit 7, we will begin our look at graph theory. The discussion will continue through Unit 8 and Unit 9, extending to trees, adjacency matrices, and a number of applications, including flow, networking, linear programming, and Matching. A graph is a construct that can be described in many ways. A graph can be described through set theory as a set *V* of vertices and set *E* of edges. Given this set definition, one can re-create the physical graph.

For example: Let the graph *G* be the set of vertices $V = \{1, 2, 3, 4, 5\}$, and let *G* be defined by the set of edges $E = \{(1, 2), (2,3), (3, 4), (3,5), (4, 5)\}$.

This graph looks like:



[D]

Many observations can be made about this graph. The first is that the graph contains a cycle as vertices 3, 4, and 5 are connected as such. Also, the graph is undirected, meaning that the edges do not require or specify direction.

A map is a great example of a graph in which the cities might be vertices and the edges might be the roads that connect them.

Graphs

While the basic definition of a graph is any set of vertices and edges, graphs can have a number of attributes. For example, a graph can have a cycle, be directed, or be connected—or not. A path of a graph, as well as the length of that path, can be determined. Graphs can be made up of subgraphs.

Applications and Representations

One can imagine the number of applications for simple graphs. In this unit, we will look at Hamiltonian cycles, the traveling salesperson problem, and shortest path algorithms. In Unit 8, we will explore specific types of graphs called trees.

Graphs, as we saw in the above example, can be represented with sets. We can also represent graphs using adjacency matrices.

A matrix is a storage construct for numbers. Example 8.5.1, on page 411, is an excellent example of a matrix being used to store a graph. In this case, the vertices of the graph are $V = \{ a, b, c, d, e\}$ and the edges are $E = \{(a,b), (a,e), (b,c), (b,e), (d,e), (d,e), (e,a), (e,b), (e,c)\}$. Figure 8.5.1, on page 411, graphs this example.

Notice that the matrix that represents the graph shows a 1 in any location where there is an edge between the two related vertices. Sometimes, there is a 2, which specifies a cycle.

Matrices are a power construct that can be added, multiplied, inverted, and so on.

Learning Activities

u07s1 - Studies

Introduction

Remember to read the introduction before starting the unit activities. This unit's introduction presents graphs, including applications and representations.

Multimedia

Click **Graph Theory** to view the presentation, which introduces this unit's topics, including:

- Edges and graphs.
- Paths and cycles.
- Hamiltonian cycles.
- A shortest-path algorithm.

Readings

Read the following in your *Discrete Mathematics* textbook:

- Chapter 8, "Graph Theory," sections 8.1 through 8.5, pages 373–414. In these sections, you will investigate paths and cycles, including the Hamiltonian cycle, the traveling salesperson problem, a shortest-path algorithm, and representations of graphs.

u07s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on graphs and adjacency matrices. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 8.1, page 382, exercises 2 and 9.
- Section 8.2, pages 392–393, exercises 2, 9, 32, and 33.
- Section 8.3, page 403, exercise 4.
- Section 8.4, page 410, exercises 3 and 8.
- Section 8.5, page 414, exercise 2.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in Unit 8, which covers all topics in Units 7 and 8, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 7** to learn a step-by-step process for solving each problem for this set.

u07d1 - Computer Networks, Distributed Systems, and Mobile Agents

Computer networks typically consist of nodes and edges—they are graphs. The nodes can typically be computers, routers, servers, and so on. The edges indicate a communication path between nodes. Assume you are a network administrator for a distributed system and that you wish to perform maintenance of all the servers in your system. You have decided to do this by using a mobile agent. To get started, look up and answer the questions in Part 1. After that, complete Part 2:

Part 1

1. What is a distributed system?
2. What is a mobile agent?

Since this agent is autonomous and moves from computer to computer, it would be smart to equip it with shortest path algorithms, so that it can go from server to server, reaching all the servers only once, in the shortest amount of time. This is the traveling sales person (TSP) problem. Note that computer networks can be modeled using graphs and that shortest paths solutions, Hamilton cycles, TSP, and so on can be found using graph-based algorithms.

Part 2

1. How would you model a computer network using a graph? What are potential nodes, edges, and weights?
2. Which of these traversal algorithms should the agent be equipped with? Which will help the agent traverse the network in an efficient manner? Explain why.
3. What is the TSP problem, and how does it relate to this question?

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Compare your response to your peer's response. Did you come up with the same results? Explain why or why not.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u07d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also

expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u07a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 8.1, page 382, problems 3, 10.
- Section 8.2, pages 392–393, problems 3, 5, 21, 29.
- Section 8.3, page 403, problem 5.
- Section 8.4, page 410, problems 2, 7.
- Section 8.5, page 414, problem 3.

u07a2 - Graph Applications and the Traveling Salesperson

In the class discussions, we have talked about how the traveling salesperson (TSP) problem and how it can be modeled using graphs. We also looked at finding a minimum length in a graph as well as Hamiltonian cycles.

Graphs, graph algorithms and methods, and graph theory are integral to IT and computer science applications and coding. For this assignment, write a two- to three-page paper that responds to each of the following questions:

1. What is a Hamiltonian cycle?

2. What is a Euler cycle?
3. What is a minimum length Hamiltonian cycle?
4. Given a graph with *n* edges, what is the time complexity of finding a Euler path? Is this a polynomial time algorithm? **Explain and show all work and the graph.** *Hint: Include the algorithm and pseudocode.*
5. Given a graph with *n* edges, can one find a minimum Hamiltonian cycle (TSP) in polynomial time? Has anyone ever proved that a polynomial time algorithm does not exist for this problem? **Explain your answers and show the graph.** *Hint: Consider NP complete problems.*
6. Offer one example of an IT or computer application that can be modeled as the TSP problem. This must be at least one paragraph.

Your calculations and work must be shown. Include references to any resources you use to complete the assignment.

Review the Graph Applications and the Traveling Sales Person Scoring Guide to understand how the assignment will be graded.

Unit 8 >> Trees

Introduction

In this unit, we will continue to talk about graphs, focusing on a specific type of graph known as a tree. A tree is a graph that has specific attributes such as a root and children of that root. In computer science, there are many applications for trees, including organizing data in a database, searching, sorting, and creating computer file systems.

Trees

Trees can be defined in the same way graphs can be defined: either using set notation or adjacency matrices. For those fond of sports, trees are often used in the definition of tournaments. There are many teams at the beginning of the tournament, each playing a specified team. The winner moves up the tree to the next level. The final winner is represented by the root or top of the tree. A binary tree has a maximum of only two children (also called nodes or vertices) attached by an edge to each parent node.

The following is a binary tree:

Adjacency Matrix

Hub 1	Hub 2	Hub 3	Hub 4	Hub 5
0	15	5	50	30
15	0	20	10	60
5	20	0	30	2
50	10	30	0	5
30	60	2	5	0

[D]

Spanning Trees and Minimal Spanning Trees

A spanning tree is found by taking any graph and determining the minimum subtree contained in the graph such that the subtree contains all the vertices of graph *G* and is connected. Page 452 of your textbook has examples, definitions, and figures that illustrate this concept.

Given a minimal spanning tree, we can also look at ways to search the tree, such as breadth-first search, depth-first search, and backtracking. These methods can be used in many applications, including gaming solutions, database searches, and shortest paths.

Prim and Kruskal

There are specific algorithms for finding minimal spanning trees. Prim's Algorithm and Kruskal's Algorithm are two such algorithms (see page 459). The pseudocode for Prim's can be found on pages 460 your course text. This algorithm is an excellent example of an algorithm type known as greedy (see page 461).

The remainder of Chapter 9 will look further into binary trees, applications, and tree traversal methods. Considerable code and examples can be found in the sections assigned in this unit's readings.

Learning Activities

u08s1 - Studies

Introduction

Begin this unit by reading the introduction, which focuses on trees, spanning trees, minimal spanning trees, Prim's algorithm, and Kruskal's algorithm.

Multimedia

Click **Trees** to view the presentation, which introduces this unit's topics, including:

- Definitions.
- Spanning trees.
- Minimal spanning trees.
- Binary trees.

Readings

Read the following in your *Discrete Mathematics* textbook:

- Chapter 9, "Trees," sections 9.1 through 9.6, pages 438–477. The sections focus on terminology and characterizations of trees, spanning trees, binary trees, and tree traversals.

u08s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on trees and tree applications. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 9.1, pages 443–445, exercises 2, 12, and 30.
- Section 9.2, pages 449–450, exercises 2, 12, and 35.
- Section 9.3, page 457, exercises 2 and 8.
- Section 9.4, page 463, exercises 2 and 6.
- Section 9.5, page 470, exercise 3.
- Section 9.6, pages 476, exercise 2.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the quiz in this unit, which covers all topics in Units 7 and 8, and for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 8** to learn a step-by-step process for solving each problem for this set.

u08d1 - Networking, Graphs, Adjacency Matrices, and Server Application

Assume you are a network administrator of a large business firm that wishes to construct an intranet. Your firm has a number of computers, servers, and services such as printers, all connected to five hubs (hubs 1, 2, 3, 4, and 5), which must now be connected to each other. Assume each hub is separated by distances shown in the following adjacency matrix:

Adjacency Matrix

Hub 1	Hub 2	Hub 3	Hub 4	Hub 5
0	15	5	50	30
15	0	20	10	60
5	20	0	30	2
50	10	30	0	5
30	60	2	5	0

[D]

Each hub must be connected to at least one other hub, therefore, you only need to make four connections to allow communication between all computers. You can further minimize the cost of the cables and time by only connecting the closest hubs.

Consider and answer the following questions to evaluate this problem:

1. What is an intranet?
2. What is an adjacency matrix, and how do they relate to graphs?
3. What is the optimal connection of the five hubs that will minimize the length of the cable required to connect them? **Note:** The length of the cable will be the distance between the hubs as shown in the adjacency matrix.
4. How did you arrive at this answer? What algorithm should be used to solve this problem?

Hint: This is a graphical minimization problem. Make sure you comment on the following: What are the vertices? What are the weights on the edges? What are we trying to minimize?

Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Compare your response to your peer's. Explain the similarities and differences between your posts.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u08d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u08a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 9.1, pages 443–445, problems 3, 16, 33.
- Section 9.2, pages 449–450, problems 3, 8, 34.
- Section 9.3, page 457, problems 3, 9.
- Section 9.4, page 463, problems 3, 8.
- Section 9.5, page 470, problem 2.
- Section 9.6, pages 476, problem 5.

u08q1 - Quiz 4

This multiple-choice quiz will gauge your understanding of the material presented in Units 7 and 8. The quiz provides an opportunity for you to demonstrate your mastery of the following course

competencies:

- Describe how discrete math theories are used in computer science.
- Apply the methodologies of discrete math.
- Apply discrete math methods and tools to solve problems encountered in a work setting.

Read the following instructions and parameters before taking the quiz:

- There is no time limit on the quiz.
- Once you have answered each question, submit your quiz to receive credit. It is automatically scored, so you will receive feedback immediately. There are 100 points possible. Each question is worth 10 points.
- You may access the quiz to review the questions after you have taken it; however, you cannot retake the quiz to change your grade. Once the grade is recorded, it cannot be changed.
- Since you can take the quiz only one time and during a single sitting, do not begin the quiz until you are ready to complete it. Do not forget to submit the quiz once you complete it.
- When you access the quiz, read the Instructions and Troubleshooting page prior to starting the quiz.
- Remember, the quiz will focus on topics directly covered in the readings and problem sets for Units 7 and 8.

Click the linked quiz title to access the quiz. If you have any issues with the quiz, contact your instructor.

Unit 9 >> Graph Applications, Flow, Networking, and Linear Programming

Introduction

In this unit, we will continue to examine graphs, trees, and their applications. Specifically, we will look at decision trees, minimum time sorting methods, network models, maximal flow algorithms, and matching. Each of these applications is directly related to IT and computer science.

Decision Trees and Minimum Time Sorting

A decision tree is a tree whose nodes, or vertices, offer a choice and which children are selected based on the choice made. Example 9.7.1 and Figure 9.7.3, on pages 477 and 478, respectively, are an excellent example set of these concepts. Computers use decision trees to reach final conclusions after interpreting a set of criteria. Some knowledge bases use decision trees to search for results.

Example: Health Knowledge Base

A health knowledge base is a type of database that contains knowledge and information about symptoms and related diagnoses. A series of questions might be posed to the user such that each response sends the user down a different path in the tree to reach the final conclusion.

Network Models

A network is a graph that can model flow. This flow can be traffic, water, waste, information, or something else. Once a network is set up, the flow can be directed, managed, and evaluated. This type of application can be used to maximize flow, minimize bottlenecks, maximize throughput, and measure time.

A maximal flow algorithm is a method for determining an optimal network and, given a network, an optimal flow path through the network. This concept is similar to understanding traffic patterns in an area and choosing a different route to your destination because it is rush hour, so traffic will be heavier in that area. Page 518 has an algorithm for determining maximal flow.

Matching

The basic matching problem is to match elements from one set to elements in another set. There are many examples of this, but consider the following IT application. Suppose you have a computer system set up with five processors and eight computer terminals. A matching problem might be to best match processors and computers while maximizing work throughput.

Learning Activities

u09s1 - Studies

Introduction

This unit's introduction will start the conversation on decision trees and minimum time sorting, network models, and matching.

Multimedia

Click **Network Models** to view the presentation, which introduces this unit's topics, including:

- Capacity.
- Flow.
- Pumping networks.
- A maximal flow algorithm.
- The max flow, min cut between.

Readings

Read the following in your *Discrete Mathematics* textbook:

- Chapter 9, "Trees," section 9.7, pages 477–483. This section investigates decision trees and the minimum time for sorting.
- Chapter 10, "Network Models," pages 506–528. Topics covered include maximal flow algorithm; the max flow, min cut theorem; and matching.

Linear Programming, Max Flow, and Matching Problems

These resources will help you learn the concepts of linear programming, max flow, and matching.

1. Read the Linear Programming, Max Flow, and Matching Problems Lecture.
2. Complete the exercises on each tab of the Linear Programming of Max Flow and Matching Problems Excel Worksheet.

u09s2 - Practice Problem Set

Practice Problems for This Unit

Complete these exercises, which focus on graph and tree applications, flow, networking, linear programming, and matching. They are located in the Exercises area of your text for the sections noted. (Note: The interactive only shows the work and answer to the problems assigned. It does not repeat the problem that is shown in the book. However, each problem is identified by the page it is on and the problem number.):

- Section 10.1, page 510, exercise 3.
- Section 10.2, page 518, exercises 6 and 8.
- Section 10.3, page 522, exercises 3 and 9.
- Section 10.4, page 527, exercises 7 and 8.

This Practice Problem Set is not graded and will not be collected. This is your opportunity to prepare for the cumulative final exam in Unit 10. To help you do well on the quizzes you are encouraged to complete the readings and problem sets, and make a note of important definitions, most of which are called out in the text. Participating in this unit's Practice Problem Set Review discussion will help you prepare, as well.

Multimedia

Click **Practice Problem Sets for Unit 9** to learn a step-by-step process for solving each problem for this set.

u09d1 - Data Storage, Retrieval, Sorting Algorithms, Time Complexity

In computer systems, the storage, retrieval, and communication of data occurs frequently and therefore must be optimized and fast. In some instances, it is best to store a list of numbers in some order (for example, in an increasing or decreasing order), to make the retrieval of some of the data very fast (for example, the maximum value).

In section 4.2 of your textbook, you read about the insertion sort algorithm (see Algorithm 4.2.3, page 180). This algorithm will sort a list of n numbers in non-decreasing order. However, in section 9.7, pages 477–483 of the text, you read about a faster sorting algorithm.

In order to fully understand this question, answer the following questions:

1. The name of the faster sorting algorithm is the tournament sort. How is this faster sorting algorithm achieved?
2. Given a list of n numbers, what is the worst case times for the insertion sort algorithm and the tournament sort algorithm?
3. In the insertion sort and the tournament sort algorithms, what discrete structures are used to store the data? Explain.
4. How does the use of the tree structure provide the framework for a tournament sort?

5. What is a divide-and-conquer algorithm? Is this a divide-and-conquer algorithm?

You may find the link in the resources section useful for this discussion. Review the Discussion Participation Scoring Guide prior to posting.

Response Guidelines

Read your peers' posts and respond to two. Compare your post with your peer's. Explain the similarities and differences.

Course Resources
Undergraduate Discussion Participation Scoring Guide
Insertion Sort

u09d2 - Practice Problem Set Review

This discussion allows you to work with your peers to complete and understand the assigned problem set for this unit. Remember, two initial posts and two response posts are **required**. Further posts are optional and recommended:

- For the first post, select a problem from this unit's problem set, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from the problem set. If there is a problem in the problem set that you are confused about, write it out fully and show any work that you have started. Note that you are stuck and ask for help. If you are able to solve every problem in the set without difficulty, post at least one other fully solved problem from the set as your second post.
- The third and fourth posts are responses. Response guidelines are provided below and in every discussion.

Take advantage of this discussion area to work together as a class on the problem set. Post as many problems as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each

person gain a better understanding of the problem set and the concepts covered in this unit. Although the problem sets are not graded they will help you prepare for the quizzes in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u09a1 - Graded Problem Set

Complete the following problems from your text. These problems are chosen to parallel the problems in The Practice Problem Sets study for this unit, so it is recommended to do those first, then post and discuss in the Practice Problem Set Review discussion so you are better prepared for this assignment.

- Section 9.7, page 482, problems 3, 13.
- Section 10.1, page 510, problem 2.
- Section 10.2, page 518, problems 5, 9.
- Section 10.3, page 522, problems 2, 8.
- Section 10.4, page 527, problems 5, 11.

u09a2 - Linear Programming, Reduction, and Max Flow Networks

In the past few units, you have learned about many discrete math and computer science optimization problems, including min cut, max flow; shortest paths; minimum spanning trees; and matching. Each of these optimization problems can be rephrased or rewritten as an equivalent linear programming problem. A linear programming problem consists of an objective (that is, a value to be maximized or minimized) and constraints.

Each of the graphical optimization problems discussed in this class can be rephrased as a linear programming problem by altering the objective or the constraints to refit the problem at hand. Another way to use a linear program to solve an optimization problem is to transform a new problem into a problem for which we already have a linear program solution—this is a reduction. The idea is that you already have a solution for a known linear programming problem. If you can somehow transform a new problem that you have into this known linear programming problem, you already have the solution. The tricky part is to figure out how to transform your problem into a known solution problem.

For this assignment, read the Notes section on page 529 of your textbook. There is a linear program on this page that describes finding a maximal flow in a network *G*, with a source *a*, a sink *z*, flows *F*, and capacities *C*. If we want to find a solution to a different problem, maybe the matching problem, for example, we could (1) formulate a linear program for the matching problem or (2) transform the matching problem into a maximal flow problem, then use the already known linear program that solves the maximal flow problem.

To better understand linear programs and reductions, write a two- to three-page paper that addresses the following questions:

1. What is a linear program? Explain in at least one paragraph. Feel free to show an example.
2. What are constraints in linear programming problems? Offer at least one example.
3. What is an optimization problem? Offer an example and explain.
4. What is a reduction (specifically, changing one problem into another)? Provide an example. You may use the Internet to locate an example.
5. Given Example 10.4.4 and Theorem 10.4.5 in the textbook, explain how you would transform the matching problem into a maximal flow problem.
6. Once you complete question 5, use the already known linear program that solves the maximal flow problem. Show all of your work and how you are doing the reduction.

Show all of your work and include references for resources you use to complete the assignment.

Hint 1: Transform the matching problem into a matching network, and then find the maximal flow to this network.

Hint 2: The goal here is to learn that, while problems might appear different, they are often the same. Knowing how to transform a new problem into a known problem can save a lot of solving

time. Use the Internet and your textbook.

Review the Linear Programming, Reduction, and Max Flow Networks Scoring Guide to understand how the assignment will be graded.

Unit 10 >> Review and Final Exam

Introduction

Welcome to Unit 10. In this unit, we will focus on preparing for the final exam. As such, no new material will be presented in this unit.

The final exam covers topics from the entire course and the questions will be much like the quiz questions.

In Unit 1, we learned about sets and set notation. A set is simply a collection of objects with order not taken into account. A graph, for example, is a set of edges and vertices. Our number system is made up of sets, some containing others. The real numbers are an infinite set of all values between negative infinity and positive infinity. We also looked at set theory and notation, as well as logic and circuits.

In Unit 2, we learned about the concepts of functions, relations, sequences, strings, and the application of relational databases. A function is a special type of relation. A function assigns or maps each value in the domain (usually known as x) to exactly one member of the range (usually known as y). A sequence is a special type of function and therefore is also a relation. For a sequence, the domain is always a set of consecutive integers, such as {1, 2, 3, 4, … }.

In Unit 3, we learned about probability theory, discrete probability theory, counting rules, permutations, and combinations. We saw that set theory and probability theory are interrelated and that an event can be thought of as a set of possible outcomes. A permutation is the number of possible ways of rearranging a discrete set (that is, one with no decimals or fractions) of objects. A combination is the number of ways one can select a set of objects when the order does NOT matter.

In Unit 4, we learned about proofs, mathematical systems, and counter examples. In previous chapters, we examined the use of logic to reach conclusions. Creating a proof required a series of logical steps that together led directly to proving the result. A direct proof assumes a set of axioms and uses logic and reasoning, with a clear sequence of steps to offer an unambiguous proof of a

claim or statement. Proof by induction uses a few specific steps, such as the basic step, the inductive step, and, finally, the conclusion.

In Unit 5, we learned about algorithms, including recursion, as well as pseudocode, control statements, and basic programming (that is, coding) principles. An algorithm is a method that will successfully complete a specific task. This unit is particularly important to the fields of IT and computer science as both use a number of different programming languages to complete tasks via algorithms.

In Unit 6, we continued our discussion of algorithms and included topics such as time complexity and cryptography, a popular computer science application.

Units 7, 8, and 9 covered graph theory, tree applications, matrix applications, and tree and graph applications for IT and computer science. Specifically, we examined decision trees, minimum time sorting methods, network models, maximal flow algorithms, and matching. Each of these applications is directly related to the fields of IT and computer science. To review, a graph is a construct that can be described in many ways. A graph can be described through set theory as a set of vertices *V* and edges *E*. A tree is a graph that has specific attributes, including a root and children of that root. In computer science, there are many applications for trees, including organizing data in a database, searching, sorting, and creating computer file systems.

In this unit, the studies and problem sets will outline what to review to prepare for the final course activities.

Learning Activities

u10s1 - Studies

The study activities in this final unit are designed to prepare you for the final exam, which will cover all of the topics introduced in this course.

Quiz Review

Reviewing the quizzes is a great way to begin preparing for the final exam. It will help you to identify the material you should review. You can access the quizzes in their respective units: 2, 4, 6, and 8.

Text Review

Review the following material in your *Discrete Math* text. The topics are listed in the order in which they were presented in the course:

- Unit 1: Chapter 1, sections 1.1 through 1.6 and Chapter 11, sections 11.1 and 11.2.
- Unit 2: Chapter 3.
- Unit 3: Sections 6.1, 6.2, 6.3, 6.5, and 6.6.
- Unit 4: Sections 2.1 through 2.4.
- Unit 5: Sections 4.1 through 4.3 and Appendix C.
- Unit 6: Chapter 4, Chapter 5, and Chapter 7.
- Unit 7: Sections 8.1 through 8.5.
- Unit 8: Sections 9.1 through 9.6.
- Unit 9: Section 9.7 and Chapter 10.

Discussion Review

Reviewing the discussions is also recommended, especially the discussions regarding problem sets. The discussion in this unit is intended for you to post any questions you have regarding the problem sets and for you to help your peers with their questions.

Assignment Review

In addition to the presentations, interactives, and problem sets in the studies, as well as the discussion, and quizzes, you may find reviewing the assignments helpful in preparing for the final exam.

u10d1 - Problem Set and Final Exam Preparation Discussion

This discussion is set up like the previous problem set discussions. In addition to using this discussion to address the unit's problem set, use the area to post any questions, concerns, or comments you have as it relates to preparing for the final exam.

- For the first post, select a problem from any problem set in this course, write it out fully, solve it fully, and post it.
- The second post can be a problem you cannot solve or another fully solved problem from any problem set. If there is a problem in any problem set that you are confused about, write it out fully and show any work you have started. Note that you are stuck and ask for help. If you were able to solve every problem without difficulty, post at least one other fully solved problem from any problem set as your second post.

- The third and fourth posts are responses. Response guidelines are provided below.

Take advantage of this discussion area to work together as a class to prepare for the final exam. Post as many problems and questions as you can and review as many of your peers' posts as you can. Ask questions. Offer answers. If you are stuck on a problem, post your question. Working as a team will help each person gain a better understanding of the problem sets and the concepts covered in this course.

Response Guidelines

The third and fourth posts, the response posts, are your chance to help your peers. Explore the posts made by your peers and find people who are stuck. Help at least two peers through the problems they are stuck on. If you are unable to locate a peer who is stuck, you may instead choose a peer post, write out your solution to the same problem, and compare solutions and methods. The goal of this area is for the entire class to work together as a group on the entire problem set. You are also expected to complete and understand the problem set on your own in preparation for the quizzes and final exam.

Course Resources
Undergraduate Discussion Participation Scoring Guide

u10q1 - Final Exam

This is the final exam. It is cumulative, covering all topics presented in the course. This exam will help gauge your skill in each of the course competencies.

Read the following instructions and parameters before taking the quiz:

- There is no time limit on the quiz.
- Once you have answered each question, submit your quiz to receive credit. It is automatically scored, so you will receive feedback immediately. There are 100 points possible. Each question is worth 4 points.
- You may access the quiz to review the questions after you have taken it; however, you cannot retake the quiz to change your grade. Once the grade is recorded, it cannot be changed.