

CSCI 0026 - DISCRETE STRUCTURES FOR COMPUTER SCIENCE

Catalog Description

Prerequisite: Completion of CSCI 12 and MATH 12 with grades of "C" or better

Hours: 72 (54 lecture, 18 laboratory)

Description: Introduction to the essential discrete structures used in Computer Science, with emphasis on their applications. Includes elementary formal logic and set theory, elementary combinatorics, recursive programming and algorithm analysis, Boolean algebra, digital logic, combinatorial circuits, graph theory, circuit design and minimization, and computer arithmetic. (C-ID COMP 152) (CSU, UC)

Course Student Learning Outcomes

- CSLO #1: Describe how formal tools of symbolic logic are used to model real-life situations, including those arising in computing contexts such as program correctness, database queries, and algorithms.
- CSLO #2: Relate the ideas of mathematical induction to recursion and recursively defined structures.
- CSLO #3: Analyze a problem to create relevant recurrence equations.
- CSLO #4: Demonstrate different traversal methods for trees and graphs.
- CSLO #5: Apply the binomial theorem to independent events and Bayes' theorem to dependent events.

Effective Term

Fall 2020

Course Type

Credit - Degree-applicable

Contact Hours

72

Outside of Class Hours

90

Total Student Learning Hours

162

Course Objectives

Lecture Objectives:

1. Convert between decimal, hexadecimal and binary representation of numbers.
2. Convert negative decimal numbers to two's complement binary form.
3. Analyze sets for complement, union and intersection.
4. Construct Venn diagrams to illustrate set relationships and operations.
5. Evaluate sets for reflexivity, symmetry, and transitivity.
6. Encrypt and decrypt a string of text using the RSA algorithm and modular arithmetic.

7. Identify problems appropriately solved with recursive functions.
8. Convert floating point notation (IEEE-754) to decimal and binary.
9. Recognize digital logic gates and their representation in graphical, algebraic, and truth table form.
10. Convert Boolean functions between algebraic expressions, truth tables and circuit diagrams.
11. Convert an English-language statement of compound propositions (including and, or, not, and quantifiers) to a symbolic form.
12. Identify and solve combinatorial word problems using permutations, combinations, pigeonhole principle, inclusion/exclusion principle, and/or product rule.
13. Compute discrete probabilities of conditional events using Bayes' Theorem.
14. Compare and contrast the "Big-O" running time of common searching and sorting algorithms.
15. Minimize circuits and Boolean expressions by creating Karnaugh Maps.
16. Analyze a graph for the shortest path between two vertices and the minimal spanning tree.
17. Create regular expressions for sets of strings.
18. Create a finite-state automaton from a written description of a set of strings.

Laboratory Objectives:

1. Design and construct logic circuits using AND, OR, NOT, NAND, and/or XOR gates from written descriptions.
2. Analyze inputs and outputs for the creation of functions that manipulate numbers, strings, lists, sets, and tuples;
3. Write and test recursive functions on numbers, lists, strings, sets, and tuples;
4. Analyze strings to derive appropriate regular expressions and test them against a corpus of text;
5. Analyze a regular language and construct a regular expression for it; and
6. Write and test functions to encrypt and decrypt strings using shift, linear, and RSA ciphers.

General Education Information

- Approved College Associate Degree GE Applicability
- CSU GE Applicability (Recommended-requires CSU approval)
- Cal-GETC Applicability (Recommended - Requires External Approval)
- IGETC Applicability (Recommended-requires CSU/UC approval)

Articulation Information

- CSU Transferable
- UC Transferable

Methods of Evaluation

- Classroom Discussions
 - Example: Example: What could be some of the consequences of discovering a fast algorithm for factoring a number into its prime factors? Grade based on participation.
- Problem Solving Examinations
 - Example: Example: Compute the 2's Complement form of -29 using an 8-bit representation. Pass/Fail Grading. Solution: Calculate the binary representation of +29: 00011101 Invert the bits: 11100010 Add 1: 11100011
- Projects
 - Example: Derive a regular expression to match ZIP codes (both 5-digit and 9-digit). The student will be provided a text

file containing sample data that includes names, addresses, and phone numbers. The regular expression should match and display all the zip codes from the file. The student solution can be checked by comparing the student output to a solution file that the instructor has created. Rubric Grading.

Repeatable

No

Methods of Instruction

- Laboratory
- Lecture/Discussion
- Distance Learning

Lab:

1. Following an instructor lecture and demonstration, students write a computer program to calculate the union and intersection of two sets. The sets can be defined in the program (for example, in Python: $A = \{3, 9, 5, 1, 2\}$, $B = \{1, 2, 3, 4, 5\}$) and the set operations applied to these sets with the results shown on the screen. Have the students work through problem sets. (Laboratory Objective 3)
2. Following an instructor lecture and demonstration, students use software such as Logisim to construct a half-adder using only NAND gates. Demonstrate on a projector, then have the students do it themselves on a computer. Compare this circuit to one that uses AND, OR, and XOR gates. Verify correct operation using a truth table and compare it to the one that Logisim can automatically generate. (Laboratory Objective 1)

Lecture:

1. Instructor lecture and demonstration on how to calculate a minimal spanning tree of a graph with at least 10 nodes and 20 edges. Draw a sample graph on the board (or use a projector) and walk through the algorithm, showing how the minimal spanning tree is derived. Hand out exercise sheets to the students with pre-drawn graphs on them and have the students apply the algorithm to the graphs. Check the graphs for correctness. (Lecture Objective 16)
2. Following an instructor lecture, students using Venn diagrams, calculate the union and intersection of two sets: for example, the set of students with blonde hair and the set of students wearing green shirts. This could be done as a kinesthetic exercise with students standing at the front of the classroom and being grouped according to their characteristics. Show how these groups of students correspond to Venn diagrams. (Lecture Objective 4)
3. Instructor lecture and demonstration on various algorithms for converting decimal numbers to binary and hexadecimal. Solicit suggestions from students for numbers to convert. After demonstrations are complete, students work through exercise sheets. (Lecture Objective 1)

Typical Out of Class Assignments

Reading Assignments

1. Read the section in the textbook on modular arithmetic to learn how to compute the multiplicative inverse. Be prepared to discuss in class.
2. Read the Wikipedia article on the RSA encryption system to learn who invented it, the underlying computations used, and its strengths and weaknesses. Be prepared to discuss in class.

Writing, Problem Solving or Performance

1. Using the "repeated division" method, convert the decimal number 29 to binary and hexadecimal. Answer: $29 / 2 = 14 \text{ r } 1$; $14 / 2 = 7 \text{ r } 0$; $7 / 2 = 3 \text{ r } 1$; $3 / 2 = 1 \text{ r } 1$; $1 / 2 = 0 \text{ r } 1$. The solution can be read in the remainders from right to left: 11101. Hexadecimal follows from the binary solution by partitioning into groups of 4 and converting each group to hexadecimal: $1 | 1101 \rightarrow 1 \text{D}$. Compute at least three strings recognized by this regular expression: $b(c|d)^*b$ Sample answers: bb, bcb, bdb, bcccd, bcdcd, bcdccdb

Other (Term projects, research papers, portfolios, etc.)

Required Materials

- Discrete Structures, Logic, and Computability
 - Author: James L. Hein
 - Publisher: Jones & Bartlett Publishers, Inc.
 - Publication Date: 2015
 - Text Edition: 4th
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Discrete Structures
 - Author: Fell, Harriet, et al
 - Publisher: Cognella Academic Publishing
 - Publication Date: 2016
 - Text Edition: 1st
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Mathematical Structures for Computer Science
 - Author: Gersting, Judith
 - Publisher: W H Freeman
 - Publication Date: 2014
 - Text Edition: 7th
 - Classic Textbook?: No
 - OER Link:
 - OER:
- Discrete Mathematics and Its Applications
 - Author: Kenneth Rosen
 - Publisher: McGraw-Hill Education
 - Publication Date: 2018
 - Text Edition: 8th
 - Classic Textbook?: No
 - OER Link:
 - OER:

Other materials and-or supplies required of students that contribute to the cost of the course.