Course Start

Chapter 1 - The Foundations: Logic and Proofs

1.7 Introduction to Proofs

In pursuit of a pythagorean proof	(KA, M)
Proof of Pythagorean Theorem	(?, V)
Origami Proof of Pythagoras	(KA, V)

1.1 - 1.6

- Introduction to Logic
- Truth Tables
- Truth Table Proofs
- Logic Laws
- Conditionals, Converses, Inverses, Contrapositives
- Quantificational Logic

1.8 Proof Methods and Strategy †

Direct Proof	(TT, V)
Proof by Case	(TT, V)
Proof by Contraposition	(TT, V)
Proof by Contradiction	(TT, V)
Mathematical Induction	(TT, V)

Chapter 2 - Basic Structures: Sets, Functions, Sequences, Sums

2.1 - Sets

Review of Sets, Notation	(B, T)
Sets	(TT, V)
Sets	(EC, V)

2.2 - Set Operations

Set Operations	(TT, V)

2.3 - Functions

Intro to Functions	(TT, V)
Onto,1-2-1,Surjective,Injective, etc.	(TT, V)
Onto,1-2-1,Surjective,Injective, etc.	(KA, V)

2.4 - Sequences and Summations

Sequences and Series	(KA, m)
Summations	(KA, ?)

Sequences and Series (EC, V)

Chapter 3 - Algorithms †

3.2 - Growth of Functions

Asymptotic Notation (KA, T)

Functions in Asymptotic Notation (KA, T)

3.3 - Complexity of Algorithms

Complexity O, Omega, Theta (X, V)

Big Theta Notation (KA, T)

Big O Notation (KA, T)

Big Omega Notation (KA, T)

Prove f(x) is Big-O of g(X) (R, V)

Prove f(x) is Big-Theta of g(X) (R, V)

Prove f(x) is Big-Omega of g(X) (R, V)

Chapter 4 - (not really covered)

4.1 Divisibility and Modular Arithmetic . . . . . .

Division Algorithm (EC, V)

GCD Euclidean Algorithm (TT, V)

4.2 Integer Representations and Algorithms . . . .

4.3 Primes and Greatest Common Divisors ....

4.3 Solving Congruences . . . . .

4.5 Applications of Congruences.....

4.6 Cryptography .....

Chapter 5 - Induction and Recursion † †

5.1 - Mathematical Induction

Induction (sum of first n integers) (KA, V)

Mathematical Induction (TT, V)

5.2 - Strong Induction and Well-Ordering

Well Ordering (TT, V)

Strong Induction Proof (DB, V)

5.3 - Recursive Definitions and Structural Induction

Recursive Algorithms (KA, T)

5.4 - Recursive Algorithms

Recursive Algorithms (KA, T)

Chapter 6 - Counting †

6.1 - Basics of Counting

Counting & Selection (EC, V)

Additional Uses For Perms and Combs (EC, V)
Counting (Probability) (KA, V)

6.2 - Pigeonhole Principle
Pigeonhole Principle (TT, V)

6.3 - Permutations and Combinations
Shortcuts (EC, V)
Permutations and Combinations (EC, V)
Permutations, Combinations, Probability (KA, V)

6.4 - Binomial Coefficients and Identities
Intro to Binomial Theorem (KA, V)
Generalizing to Binomial Theorem (KA, V)

MIDTERM

Chapter 2 - Basic Structures: Sets, Functions, Sequences, Sums
2.5 - Cardinality of Sets (countability of infinite sets) †

Chapter 4
4.1 Divisibility and Modular Arithmetic . . . . . .
Division Algorithm (EC, V)
GCD Euclidean Algorithm (TT, V)

Chapter 6 - Counting (continued) †
6.5 - Distribution & Derangement (EC, V)

Chapter 8 - Advanced Counting Techniques †
8.1 - Applications of Recurrence Relations
Recurrence Relations (TT, V)
Recurrence Relations (KA, V)
8.2 - Solving Linear Recurrence Relations
Homogeneous Recurrence Relations (TT, V)
Non-Homogeneous Recurrence Relations (TT, V)

Chapter 9 - Relations †
9.1 - Relations and Their Properties
Intro to Relations (TT, V)

9.3 - Representing Relations
9.5 - Equivalence Relations

Chapter 10 - Graphs (no Dijkstra) † †
10.1 - Graphs and Graph Models
Graph Representation and describing graphs (KA, T)
Representing Graphs (KA, T)
10.2 - Graph Terminology and Special Types of Graphs
Graph Terminology (TT, V)
Vertex Degree and Regular Graphs (TT, V)
10.3 - Representing Graphs and Graph Isomorphism
Isomorphism and Bipartite Graphs (TT, V)
10.4 - Connectivity
Subgraphs, Complements, and Complete Graphs (TT, V)
10.5 - Euler and Hamilton Paths
Euler Circuits / Paths (TT, V)
Hamilton Paths (TT, V)
10.6 - Shortest Path (not covered)
10.7 - Planar Graphs
Proof: Euler's Equation (e-v+2=r) (3B, V)
Euler's Theorem
(TT, V)
Planar Graph (?, V)
Planar Graph (TT, V)

Chapter 13 - Modeling Computation †
Languages, Grammars, and FSMs
13.1 - Languages and Grammars
Formal Languages
Formal Language Examples
13.2 - Finite-State Machines with Output
albert Finite State Machines
Formal Definition of a Finite State Machine
13.3 - Finite-State Machines with No Output

Final

Chapters are from Rosen (PDF, Amazon). There is another book also, Epp (PDF, Amazon)

Sources = {
    KA = Khan Academy *
    R = Randerson
    TT = TrevTutor
    B = Berkeley CS70
    X = xoax.net
    DB = Math Doctor Bob
    EC = Endeavor Careers *
    3B = 3Blue1Brown
}

Methods = {
    V = Video
    T = Text
    M = Mixed Media
}
* = Very Clear
† = Likely exam question