

From Foundations to Research: An Algorithms Roadmap

Purpose and Scope. This roadmap is designed for undergraduate students and early graduate learners who want to develop a rigorous, proof-oriented understanding of algorithms and algorithmic thinking. Rather than serving as a checklist of topics, the goal is to build strong modeling intuition, formal reasoning skills, and the ability to design and analyze algorithms across paradigms such as divide-and-conquer, greedy methods, dynamic programming, graph algorithms, and complexity theory.

This roadmap reflects the path I personally followed to build a strong foundation in algorithms. While there are many excellent resources and valid approaches to learning algorithms, this structure emphasizes depth, conceptual clarity, and proof-based reasoning. Readers are encouraged to adapt the roadmap based on their background, goals, or preferred resources, while preserving the core progression of ideas and algorithmic paradigms.

Following this roadmap prepares the reader to confidently approach advanced topics such as randomized algorithms, approximation algorithms, streaming and online algorithms, and theoretical computer science research. Upon completion, readers are well-positioned to engage with graduate-level algorithms courses, research papers, and independent problem-solving in algorithms and complexity theory.

References Used. Throughout the roadmap, the following standard resources are referenced:

- **CLRS:** *Introduction to Algorithms* by Cormen, Leiserson, Rivest, and Stein
 - **KT:** *Algorithm Design* by Kleinberg and Tardos
 - **Roughgarden Courses:** Tim Roughgarden's *Algorithms Specialization* on Coursera
-

Phase 0: Foundations

0.1 Algorithmic Analysis Core

- **CLRS Chapters 1–3:** Role of algorithms, asymptotic analysis, running times
- **Roughgarden Course 1:** Week 1

0.2 Sorting & Selection (Canonical Problems)

- **CLRS Chapter 6:** Heapsort
- **CLRS Chapter 7:** Quicksort
- **CLRS Chapter 8:** Sorting in linear time
- **CLRS Chapter 9:** Medians and order statistics

0.3 Core Data Structures

- **CLRS Chapter 10:** Elementary data structures
- **CLRS Chapter 11:** Hash tables
- **CLRS Chapter 12:** Binary search trees
- **CLRS Chapter 13:** Red-black trees
- **Roughgarden Course 2:** Week 3, Week 4

0.4 Amortization & Union-Find

- **CLRS Chapter 16:** Amortized analysis
 - **CLRS Chapter 19:** Disjoint sets
-

Phase 1: Core Algorithmic Design (KT → CLRS)

1.1 Algorithmic Mindset & Modeling

- **KT Chapters 1–2:** Stable matching, problem modeling, tractability

1.2 Divide and Conquer

- **KT Chapter 5:** Design intuition and recurrences
- **CLRS Chapter 4:** Formal divide-and-conquer analysis
- **Roughgarden Course 1:** Week 2, Week 3

1.3 Greedy Algorithms

- **KT Chapter 4:** Exchange arguments, MST, Huffman coding
- **CLRS Chapter 15:** Greedy-choice property, formal proofs
- **Roughgarden Course 3:** Week 1, Week 2

1.4 Dynamic Programming

- **KT Chapter 6:** DP modeling and problem decomposition
 - **CLRS Chapter 14:** Formal DP framework
 - **Roughgarden Course 3:** Week 3, Week 4
-

Phase 2: Graphs & Flows (KT → CLRS)

2.1 Graph Algorithms

- **KT Chapter 3:** Graph traversal and connectivity
- **CLRS Chapter 20:** BFS, DFS, SCCs
- **CLRS Chapter 21:** Minimum spanning trees
- **CLRS Chapter 22:** Single-source shortest paths
- **Roughgarden Course 2:** Week 1, Week 2

2.2 Network Flow

- **KT Chapter 7:** Max-flow, min-cut, applications
 - **CLRS Chapter 24:** Formal flow algorithms
-

Phase 3: Limits of Computation

3.1 NP-Completeness

- **KT Chapter 8:** Reductions, gadgets, NP intuition
 - **CLRS Chapter 34:** Formal NP-completeness proofs
 - **Roughgarden Course 4:** Week 1, Week 2
-

Phase 4: Research-Oriented Topics

4.1 Randomized Algorithms

- **KT Chapter 13:** Randomized algorithms and Chernoff bounds
- **Roughgarden Course 1:** Week 4

4.2 Approximation Algorithms

- **KT Chapter 11:** Greedy, LP rounding, PTAS
- **CLRS Chapter 35:** Approximation techniques
- **Roughgarden Course 4:** Week 3

4.3 Optional CLRS Topics (On Demand)

- **CLRS Chapter 23:** All-pairs shortest paths
- **CLRS Chapter 32:** String matching
- **CLRS Chapter 30:** FFT
- **CLRS Chapter 29:** Linear programming
- **Roughgarden Course 4:** Week 4