📚 COMPLETE_SYLLABUS_v9.2_FINAL.md — BEST OF BOTH v8.0 & v9.0

Version: 9.2 (Unified Complete Curriculum)
Date: December 30, 2025
Status: ☑ OFFICIAL FINAL SYLLABUS
Total Duration: 16.75 weeks | 85 days | 75+ topics | 460,000-570,000+ words

---

🎯 CURRICULUM OVERVIEW

Total Structure

```
Weeks 1-4:      Foundations & Core Patterns (20 topics)
Week 4.5:       ⭐ TIER 1 - Critical Patterns (6 topics, 70-80% coverage)
Week 4.75:      ⭐ TIER 1.5 - String Manipulation Patterns (4 topics) ✨ NEW in
v9.0
Weeks 5-8:      Trees, Graphs & Specialized (24 topics)
Week 5.5:       🟢 TIER 2 - Strategic Patterns (3 topics, 80-88% coverage)
Weeks 9-12:     Advanced & Mastery (22 topics)
Week 13:        🟢 TIER 3 - Extensions (7 topics, 85-95% coverage)
Weeks 14-16:    Advanced Mastery & Deep Dives (15+ topics)
```

🆕 **v9.2 UNIFIED FEATURES:**

- ☑ **Week 4.75 (String Patterns):** Restored from v9.0 (adds 15% interview coverage).
- ☑ **Detailed Breakdown:** Using v8.0's comprehensive bullet-point style.
- ☑ **Restored Topics:** All critical restorations from v8.0 (Merge Intervals Adv, Cyclic Sort, etc.) retained.
- ☑ **Modern Additions:** All v9.0 additions (Reservoir Sampling, Top K) retained.

📖 WEEK-BY-WEEK DETAILED SYLLABUS

WEEK 1: FOUNDATIONS I — Computational Fundamentals

Goal: Understand how computers work and measure algorithm efficiency
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟢 Easy/Fundamental
Prerequisites: None
Interview Coverage: 0% (foundation only)

Daily Breakdown

Day 1: RAM Model & Pointers

- Memory layout & RAM model
- CPU cache (L1, L2, L3)
- Pointer arithmetic & dereferencing
- Virtual memory & paging (TLB)
- Real system examples (Linux kernel, JVM)

Day 2: Asymptotic Analysis (Big-O)

- Big-O, Big-Omega, Big-Theta notation
- O(1), O(log n), O(n), O(n log n), $O(n^2)$, $O(2^n)$
- Amortized analysis
- Master theorem
- Common misconceptions & complexity classes

Day 3: Space Complexity

- Auxiliary space vs total space
- Stack vs heap allocation
- Memory overhead (pointers, object headers)
- Space-time trade-offs
- Practical considerations

Day 4: Recursion I

- Call stack mechanics
- Base cases & recursive cases
- Stack overflow risks
- Recursion vs iteration
- Proof by induction intuition

Day 5: Recursion II

- Advanced recursion patterns
- Tail recursion optimization
- Mutual recursion
- Recursion with memoization
- Design patterns

WEEK 2: FOUNDATIONS II — Linear Data Structures

Goal: Master fundamental linear data structures
Duration: 5 days | ~25,000-35,000 words
Difficulty: Easy
Prerequisites: Week 1
Interview Coverage: 10-15%

Daily Breakdown

Day 1: Arrays

- Static arrays & indexing
- Cache locality & memory layout
- Insertion & deletion costs
- Array variants (jagged, multidimensional)
- Real systems (Python lists, Java arrays)

Day 2: Dynamic Arrays

- Amortized analysis (doubling strategy)
- Growth strategies & resizing costs
- Push/Pop operations
- Resizable array implementations (ArrayList, std::vector)

Day 3: Linked Lists

- Singly linked lists mechanics
- Memory layout & cache misses
- Insertion & deletion operations
- Doubly & circular linked lists
- Real systems usage

Day 4: Stacks & Queues

- LIFO vs FIFO semantics
- Stack applications (RPN, undo/redo, DFS)
- Queue applications (BFS, scheduling)
- Circular queues (Ring Buffer)
- Priority queues intro

Day 5: Binary Search

- Binary search prerequisites (sorted)
- Complexity O(log n)
- Edge cases & off-by-one errors
- Search variants (exact, lower bound, upper bound)
- Applications in rotated arrays

WEEK 3: FOUNDATIONS III — Sorting & Hashing

Goal: Master sorting algorithms and hash tables
Duration: 5 days | ~25,000-35,000 words
Difficulty: ◍ Medium
Prerequisites: Week 1-2
Interview Coverage: 15-25%

Daily Breakdown

Day 1: Elementary Sorts

- Bubble Sort, Selection Sort, Insertion Sort

- $O(n^2)$ analysis & cache behavior
- Stable vs unstable sorting
- When to use elementary sorts (small N)
- Real systems (TimSort initial pass)

Day 2: Merge Sort & Quick Sort

- Merge Sort: O(n log n) divide-and-conquer
- Quick Sort: O(n log n) average, $O(n^2)$ worst
- Partitioning strategies (Lomuto, Hoare)
- In-place variants
- Real systems (Python Timsort, Java sort)

Day 3: Heap Sort & Variants

- Heap structure & properties
- Heap Sort: O(n log n) in-place
- Heapify operations
- Bottom-up vs top-down construction
- Real systems (priority queues, OS scheduling)

Day 4: Hash Tables I

- Hash function design
- Collision resolution (chaining, probing)
- Load factors & resizing strategies
- Hash table anatomy
- Real systems (HashMap, Redis hashes)

Day 5: Hash Tables II

- Open addressing strategies
- Cuckoo hashing, Robin Hood hashing
- Perfect hashing
- Universal hashing
- Real systems (Python dict, Java HashMap)

WEEK 4: PROBLEM-SOLVING PATTERNS

Goal: Learn systematic techniques for solving problems
Duration: 5 days | ~25,000-35,000 words
Difficulty: ◍ Medium
Prerequisites: Week 1-3
Interview Coverage: 25-40%

Daily Breakdown

Day 1: Two Pointers

- Two-pointer pattern mechanics

- Container with most water
- Merge sorted arrays
- Partition algorithms
- Pattern variations (same direction vs opposite)

Day 2: Sliding Window (Fixed)

- Fixed-size sliding window mechanics
- Maximum/minimum in window
- Average of all subarrays
- Repeated character longest substring (fixed)
- Optimization strategies

Day 3: Sliding Window (Variable)

- Variable-size sliding window
- Longest substring with constraints
- Minimum window substring
- Character frequency problems
- Two-pointer vs sliding window distinction

Day 4: Divide and Conquer Pattern

- Divide and conquer strategy (General)
- Merge Sort & Quick Sort (from Week 3 context)
- Strassen's matrix multiplication
- Master Theorem application
- **Interview Frequency:** 25-30%

Day 5: Binary Search as Problem-Solving Pattern

- Binary search on value range (not array)
- Binary search on answer space
- Finding boundaries (first/last occurrence)
- Rotated array problems
- **Interview Frequency:** 70% (Very High)

⭐ WEEK 4.5: TIER 1 — CRITICAL PROBLEM-SOLVING PATTERNS

Goal: Master patterns that solve 70-80% of interview problems
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟡 Medium (Critical!)
Prerequisites: Week 1-4
Interview Coverage: 70-80% cumulative

Daily Breakdown

Day 1: Hash Map / Hash Set Patterns

- HashMap: O(1) average lookup/insertion

- Two sum problem
- Anagram detection
- Frequency counting
- Real-world: LRU cache foundation

Day 2: Monotonic Stack

- Maintaining monotonic order
- Next greater element
- Trapping rain water
- Largest rectangle in histogram
- Daily temperatures problem

Day 3: Merge Operations & Intervals

- Merging sorted arrays/lists (O(n) / O(m+n))
- Merge K sorted lists (Heap approach)
- **Merge Intervals:** Scheduling, overlapping intervals
- Insert Interval
- Real systems (calendar scheduling)

Day 4 (Part A): Partition & Cyclic Sort

- Dutch National Flag problem (0, 1, 2 sort)
- Move zeroes to end
- **Cyclic Sort pattern:** Finding missing numbers, duplicates
- In-place segregation O(1) space

Day 4 (Part B): Kadane's Algorithm

- Maximum subarray problem
- Maximum product subarray
- DP formulation
- Constraint variations (circular)
- Real-world: financial analysis

Day 5: Fast & Slow Pointers

- Fast & Slow mechanics
- Linked list cycle detection
- Finding cycle start
- Midpoint finding / Happy Number
- **Interview Frequency:** 60%

⭐ WEEK 4.75: TIER 1.5 — STRING MANIPULATION PATTERNS ✨ (NEW in v9)

Goal: Achieve 85-90% string interview coverage
Duration: 4 days | ~15,000-20,000 words
Difficulty: 🟨 Medium

Prerequisites: Week 4.5
Interview Coverage: +10-15% (High ROI)

Daily Breakdown

Day 1: Palindrome Patterns

- Two-pointer expansion (Expand Around Center)
- Longest Palindromic Substring
- Valid Palindrome variants
- Palindrome partitioning
- **Interview Frequency:** 25-30%

Day 2: Substring & Sliding Window on Strings

- Longest Substring Without Repeating Characters
- Character Replacement (Longest Repeating Character Replacement)
- Permutation in String (Anagrams)
- Minimum Window Substring (String focus)
- **Interview Frequency:** 30-35%

Day 3: Parentheses & Bracket Matching

- Valid Parentheses (Stack pattern)
- Generate Parentheses (Backtracking)
- Longest Valid Parentheses (DP/Stack)
- Minimum Remove to Make Valid
- **Interview Frequency:** 15-20%

Day 4: String Transformations & Building

- String to Integer (atoi)
- Integer to Roman / Roman to Integer
- Zigzag Conversion
- String compression / RLE
- StringBuilder usage & performance
- **Interview Frequency:** 15-20%

WEEK 5: TREES & HEAPS

Goal: Master hierarchical data structures
Duration: 5 days | ~25,000-35,000 words
Difficulty: ◍ Medium
Prerequisites: Week 4.75
Interview Coverage: 35-50%

Daily Breakdown

Day 1: Binary Tree Anatomy

- Terminology (root, leaf, height)
- Properties & types (perfect, complete, full)
- Representation (array vs linked)
- Real systems (heap storage, DOM)

Day 2: Tree Traversals

- In-order, pre-order, post-order (DFS)
- Level-order traversal (BFS)
- Morris traversal (O(1) space)
- Recursion vs iteration

Day 3: Binary Search Trees

- BST properties & invariants
- Search, insertion, deletion
- Balanced vs unbalanced
- AVL/Red-Black intro
- Real systems (Databases indices)

Day 4: Heaps & Top K Elements Pattern

- Heap structure (Min/Max)
- Heapify operations
- **"Top K Elements" Pattern:** Kth largest, K most frequent
- Streaming data (Top K)
- Real systems (Task scheduling)

Day 5: Balanced Trees

- AVL rotations
- Red-Black tree properties
- Self-balancing mechanisms
- Performance guarantees

WEEK 5.5: 🟠 TIER 2 — STRATEGIC PATTERNS

Goal: Master advanced patterns for "Strong Hire"
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟠 Medium (Strategic)
Prerequisites: Week 5
Interview Coverage: 80-88% cumulative

Daily Breakdown

Day 1: Difference Array

- Difference array technique
- Range updates O(1)
- Range query O(1)

- 2D difference arrays
- Event processing applications

Day 2: In-Place Transformations

- Rearranging arrays in-place
- Rotate matrix
- Spiral matrix traversal
- Matrix transpose

Day 3: Advanced String Patterns

- Manacher's algorithm (palindromes)
- Z-algorithm
- KMP advanced variants
- String hashing (Rabin-Karp recap)

WEEK 6: GRAPHS I — FOUNDATIONS

Goal: Master graph representations and basic traversals
Duration: 5 days | ~25,000-35,000 words
Difficulty: ▨ Medium
Prerequisites: Week 5
Interview Coverage: 25-40%

Daily Breakdown

Day 1: Graph Representations

- Adjacency matrix vs list
- Edge list
- Implicit graphs
- Memory trade-offs

Day 2: Breadth-First Search (BFS)

- Queue-based traversal
- Shortest path (unweighted)
- Connected components
- Applications (social networks)

Day 3: Depth-First Search (DFS)

- Stack-based traversal
- Recursion & iteration
- Applications (maze solving, puzzle solvers)

Day 4: Graph Cycles & Connectivity

- Cycle detection (directed/undirected)

- Connected components
- Strongly connected components (SCC)
- Union-Find for connectivity

Day 5: Shortest Path I

- Dijkstra's algorithm
- Priority queue implementation
- Non-negative weights
- Real systems (GPS, routing)

WEEK 7: GRAPHS II — ADVANCED

Goal: Master advanced graph algorithms
Duration: 5 days | ~25,000-35,000 words
Difficulty: Hard
Prerequisites: Week 6
Interview Coverage: 30-45%

Daily Breakdown

Day 1: Shortest Path II

- Bellman-Ford algorithm
- Floyd-Warshall (All-pairs)
- Negative weight cycles
- Routing protocols

Day 2: Minimum Spanning Trees

- Kruskal's algorithm
- Prim's algorithm
- Cut property & cycle property
- MST applications (clustering)

Day 3: Topological Sort

- DAG properties
- DFS-based sort
- Kahn's algorithm (BFS-based)
- Build systems / Course schedule

Day 4: Network Flow I

- Flow concepts
- Ford-Fulkerson algorithm
- Max-flow min-cut theorem
- Applications (matching)

Day 5: Network Flow II

- Edmonds-Karp algorithm
- Bipartite matching
- Flow decomposition
- Real-world applications

WEEK 8: SPECIALIZED DATA STRUCTURES

Goal: Master specialized structures for specific domains
Duration: 5 days | ~25,000-35,000 words
Difficulty: ▨ Medium
Prerequisites: Week 7
Interview Coverage: 15-25%

Daily Breakdown

Day 1: Tries

- Prefix trees
- Insert, search, delete
- Autocomplete implementation
- Real systems (Spell checkers, IP routing)

Day 2: Segment Trees

- Construction
- Range query operations
- Point updates
- Lazy propagation

Day 3: Fenwick Trees

- Binary Indexed Tree
- Efficient range sums
- Point updates O(log n)
- Trading systems

Day 4: Union-Find / Disjoint Set

- Path compression
- Union by rank/size
- Kruskal's MST application
- Image processing (connected components)

Day 5: Suffix Structures

- Suffix arrays & trees
- Pattern matching
- Bioinformatics applications

WEEK 9: STRING & MATH MASTERY

Goal: Master string algorithms and mathematical foundations
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟨 Medium-Hard
Prerequisites: Week 8
Interview Coverage: 15-30%

Daily Breakdown

Day 1: String Matching (KMP)

- Pattern matching basics
- KMP (Knuth-Morris-Pratt)
- Failure function
- Linear-time matching

Day 2: String Matching (Rabin-Karp)

- Rolling hash technique
- Polynomial hashing
- Multi-pattern matching

Day 3: Number Theory & Bit Manipulation

- **Bit Manipulation:** XOR tricks, subset generation, bitmasks
- **Number Theory:** Primes, GCD/LCM, Modular arithmetic
- Interview frequency: 15%

Day 4: Modular Arithmetic & Probability

- Extended GCD, CRT
- **Reservoir Sampling:** Streaming data sampling
- Probability basics
- Interview frequency: 8%

Day 5: Computational Geometry

- Point, line, polygon basics
- Convex hull (Graham scan)
- Line intersection
- GIS applications

WEEK 10: GREEDY & BACKTRACKING

Goal: Master optimization and exhaustive search
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟨 Medium-Hard
Prerequisites: Week 9
Interview Coverage: 20-35%

Daily Breakdown

Day 1: Greedy Algorithms

- Greedy choice property
- Optimal substructure
- Activity selection, Huffman coding
- Greedy vs DP

Day 2: Backtracking I

- Decision trees
- Permutations, combinations
- N-queens, Subsets

Day 3: Meet-in-the-Middle (Optional)

- O(2^n) to O(2^(n/2))
- Two-pass algorithm
- Subset sum variant

Day 4: Backtracking II

- Constraint satisfaction
- Sudoku solver, Graph coloring
- State space pruning

Day 5: Backtracking III

- Advanced techniques
- Word ladder, Boggle
- Parallelization

WEEK 11: DYNAMIC PROGRAMMING MASTERY

Goal: Master optimization through memoization and tabulation
Duration: 5 days | ~25,000-35,000 words
Difficulty: Hard
Prerequisites: Week 10
Interview Coverage: 30-45%

Daily Breakdown

Day 1: DP Fundamentals

- Optimal substructure & Overlapping subproblems
- Memoization vs Tabulation
- State representation

Day 2: 1D DP & Classic Problems

- Climbing stairs, House robber
- Longest increasing subsequence (LIS)
- Coin change

Day 3: 2D/Sequence DP

- 0/1 Knapsack
- Longest common subsequence (LCS)
- Edit distance
- Matrix chain multiplication

Day 4: Advanced DP Techniques

- Digit DP
- Bitmask DP
- DP on Trees
- State compression

Day 5: DP Optimizations

- Space optimization
- Convex Hull Trick (CHT)
- Knuth optimization
- Divide and conquer optimization

WEEK 12: INTERVIEW MASTERY & INTEGRATION

Goal: Solve complex problems by integrating concepts
Duration: 5 days | ~25,000-35,000 words
Difficulty: Hard
Prerequisites: Week 11
Interview Coverage: 85-92%

Daily Breakdown

Day 1: Merge Intervals (Advanced)

- Interval trees
- Scheduling problems
- Overlapping detection

Day 2: Monotonic Stack (Advanced)

- Next smaller/greater variants
- Stock span, Trapping water II
- Circular array variations

Day 3: Cyclic Sort Pattern

- Finding missing/duplicate numbers

- First missing positive
- In-place permutation

Day 4: Matrix Problems (Advanced)

- Matrix traversal patterns
- Set matrix zeroes
- 2D Search
- Rotate/Spiral recap

Day 5: System Integration

- Multi-concept problems
- Algorithm selection strategies
- Time/space optimization
- Real interview simulation

WEEK 13: TIER 3 — ADVANCED EXTENSIONS

Goal: Master specialized algorithms (Elite Tier)
Duration: 7 days | ~35,000-45,000 words
Difficulty: Hard (Specialist)
Prerequisites: Week 12
Interview Coverage: 85-95%

Daily Breakdown

Day 1: Fast & Slow Pointers (Extended)

- Partition list
- Reorder list
- Advanced cycle problems

Day 2: Reverse & Two Pointers

- String reversal patterns
- Reverse words
- Array to deque conversion

Day 3: Matrix Traversal (Advanced)

- Diagonal traversal
- Boundary traversal
- Advanced spiral

Day 4: Matrix Exponentiation

- Binary lifting
- Fibonacci via matrix
- Linear recurrence relations

Day 5: Union-Find Advanced

- Weighted Union-Find
- Path compression optimizations
- Kruskal's integration

Day 6: Conversion & Encoding

- Run-length encoding (RLE)
- String compression techniques
- Decoding variations

Day 7: Advanced Optimization

- Ternary search
- Parallel algorithms
- Approximation algorithms

WEEK 14: ADVANCED MASTERY — DEEP DIVES (PART 1)

Goal: Expert level specialization
Duration: 5 days | ~25,000-35,000 words
Difficulty: Expert
Prerequisites: Week 13
Interview Coverage: 90-95%

Daily Breakdown

Day 1: Segment Trees Advanced

- Custom operators, 2D trees
- Persistent segment trees

Day 2: Heavy-Light Decomposition

- Tree path decomposition
- Dynamic tree queries

Day 3: Advanced Graph Algorithms

- Tarjan's SCC
- 2-SAT
- Biconnected components

Day 4: Advanced DP Optimizations

- CHT 2D
- Monotone queue optimization

Day 5: String Algorithms (Advanced)

- Aho-Corasick
- Suffix array construction
- Longest palindrome (Manacher's recap)

WEEK 15: ADVANCED MASTERY — DEEP DIVES (PART 2)

Goal: Expert level algorithms
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟤 Expert
Prerequisites: Week 14
Interview Coverage: 92-95%

Daily Breakdown

Day 1: Advanced Hash Structures

- Bloom filters
- Count-min sketch
- HyperLogLog

Day 2: Advanced Graph Coloring

- K-coloring
- Chromatic polynomial

Day 3: Advanced Network Flow

- Min-cost max-flow
- Circulation with demands

Day 4: Advanced Geometry

- Delaunay triangulation
- Voronoi diagrams

Day 5: System Design Patterns

- Scalability
- Problem integration
- Design interviews

WEEK 16: MOCK INTERVIEWS & FINAL MASTERY

Goal: Final preparation
Duration: 5 days | ~25,000-35,000 words
Difficulty: 🟤 Expert
Prerequisites: Week 15
Interview Coverage: 95%+

Daily Breakdown

Day 1: Mock Interview Session 1 (60-90 min, 2-3 problems)
Day 2: Mock Interview Session 2 (Different types)
Day 3: Weak Points Review (Targeted practice)
Day 4: System Integration Performance (Complex problems)
Day 5: Final Preparation (Rapid fire, strategies)

☑ QUALITY STANDARDS (v9.2)

- **Structure:** 11 Sections + Cognitive Lenses + Supplementary Outcomes
- **Format:** Markdown (No LaTeX), C# Code (Only if needed)
- **Visuals:** Mermaid diagrams preferred, ASCII backup
- **Quantity:** 5-7 instructional files/week, 6-7 support files/week
- **Volume:** 30,000-35,000 words/week

Generated: December 30, 2025
Version: 9.2 (Unified Complete)
Status: ☑ OFFICIAL FINAL SYLLABUS