Coding Streak Framework

Phase 1 — Syntax & Fluency Foundation

Focus:

- loops
- conditionals
- simple array traversal
- string manipulation
- math problems
- basic brute-force thinking
- time complexity intuition ($O(n)$, $O(n^2)$)

Goal: Students code comfortably without syntax hesitation.

Phase 2 — Core Logic & Counting Thinking

Focus:

- prefix sum
- counting logic
- hashing basics (frequency map)
- pattern observation problems
- basic set/map usage

Goal: Students learn **logic construction before writing loops**.

Phase 3 — Algorithmic Thinking & Classical Algorithms

Focus:

- two pointer
- sliding window
- basic recursion
- greedy basics
- searching algorithms
    - linear search
    - binary search
- sorting algorithms
    - bubble, selection, insertion
    - merge, quick, heap
    - counting and radix
    - bucket

Goal: Students start thinking **approach-first, code-next**.

Phase 4 — Linear Data Structures

Focus:

- linked list
- stack
- queue
- circular queue
- monotonic stack
- priority queue
- heap implementation

Goal: Comfort with **interview-classic linear DS problems**.

Phase 5 — Non-Linear Data Structures

Focus:

- binary tree, binary search tree, AVL
- Red-Black (conceptual understanding + limited coding)
- trie
- graph basics (representation, BFS, DFS)

Goal: Students become comfortable with recursion-based structures.

Phase 6 — Advanced Algorithms & Placement Level Thinking

Focus:

- greedy advanced
- dynamic programming (core patterns)
- backtracking
- divide and conquer
- string matching algorithms
  - KMP
  - Rabin-Karp
- graph algorithms
  - shortest path
  - MST
  - topological sort

Goal: Placement-level problem solving ability.