

Rules That Keep This Roadmap Mine

AUTHOR Mohammed Efaz | IDEA Roadmap | WORDS 988 | READ TIME ~6 min

Rules that keep this roadmap mine

This file protects the decisions I made while researching the roadmap. Anyone changing the curriculum, including a future version of me, needs to read it first.

What I am trying to build

I want deep practical ability across:

- C and modern C
- Python data structures, algorithms, and LeetCode
- Linux command-line use, shell scripting, internals, system programming, and kernel programming
- modern C++ and asynchronous systems
- bare-metal embedded C
- Kali Linux, ethical hacking, and Linux hardening
- machine learning and deep learning
- LLM construction
- CUDA, GPU-kernel work, and AI inference and performance engineering
- reasoning models, reinforcement learning, and RLHF

Finishing books is not the goal by itself. I want competence backed by implementation, testing, debugging, profiling, and real lab work.

The choices I do not want quietly reversed

1. Teaching style matters to me

I learn best from clear, progressive, practical books with explanations, examples, exercises, and projects. K. N. King's *C Programming: A Modern Approach* is the closest model for what works for me.

Prestige does not cancel fit. If I dislike the way a famous book teaches, I am not going to make it mandatory because other people treat it as a rite of passage.

2. Computer Systems: A Programmer's Perspective is not mandatory

I dislike the presentation style of *Computer Systems: A Programmer's Perspective*, and I do not intend to use it as a core textbook.

It must not return as a gate for Stage 3, Linux, C++, kernel work, CUDA, or performance engineering. It remains in the library inventory as an unselected book. I accept that leaving it out creates a gap in formal computer-architecture coverage unless I later choose a different resource.

3. Python is my main DSA and LeetCode language

My DSA route is:

1. *A Common-Sense Guide to Data Structures and Algorithms in Python, Volume 1*
2. *A Common-Sense Guide to Data Structures and Algorithms in Python, Volume 2*
3. LeetCode 75 in Python

C is my main systems language, but that does not mean every subject needs to be learnt through C. I chose Python for DSA so I can focus on the algorithms without mixing in manual memory management.

4. Algorithmic Thinking is optional

I may read *Algorithmic Thinking, 2nd Edition* after Stage 2 if I want more C-based problem-solving practice.

It is not my main DSA book, a stage requirement, a replacement for the 2 Wengrow volumes, or a blocker for Stage 3. I rejected it as the primary text because its selective, problem-first structure does not match the way I want to learn the subject.

5. Owning a book does not make it part of the plan

My Calibre library is a pool of candidates. I choose from it based on:

- relevance to the goal,
- prerequisite order,
- teaching-style fit,
- unique value,
- overlap,
- and whether I am likely to finish the book and practise what it teaches.

I do not need to force every computer-science title I own into the default route.

6. Security and reasoning stay in the main roadmap

I want both offensive and defensive Linux security. I also want reasoning-model and post-training work.

I will not push either subject into an optional list merely to reach CUDA or inference performance sooner.

7. I keep 1 roadmap

This repository contains 1 default curriculum. Git preserves older versions, so parallel active and archived roadmap folders only create confusion.

I will not create another competing roadmap directory unless I deliberately decide that I need one.

8. I do not need artificial lanes

Inside the current stage, I can work through 2 or 3 books and choose based on my energy. The stage order protects the prerequisites, but I do not need permanent lanes, percentage splits, daily page quotas, or equal-progress rules.

9. Different books need different reading modes

I will not force every resource into cover-to-cover reading.

- I will follow the main teaching sequence and practise when I use a textbook.
- I will build the projects when I use a project book.
- I will make a structured pass through manuals and pocket guides, then keep them for lookup.
- I will read lighter narrative books normally.
- I will leave advanced research texts until I have the stated prerequisites.

10. New books need a reason

If I find a real gap later, I can add a resource. Before I do, I need to:

1. name the missing capability,
2. show why the current library does not cover it,
3. propose the smallest useful addition,
4. check the latest edition,
5. and make a deliberate decision rather than adding it on impulse.

11. This repository is only for the learning plan

This repository holds the learning journey, curriculum, library decisions, practice rules, and current position. Content production, branding, publishing plans, and unrelated project management belong elsewhere.

My verified position

- Active stage: Stage 1.
- Active anchor: *C Programming: A Modern Approach, 2nd Edition*.
- Current position: Chapter 6, Loops.
- Completed work: all 12 Chapter 6 programming projects are implemented and verified.
- Current companion: *The Linux Command Line, 3rd Edition*.
- Current DSA book: Wengrow Volume 1 in Python.
- Lighter reading: *UNIX: A History and a Memoir*.

Checklist before I change the route

- ☐ I read this file.
- ☐ I did not treat book ownership as a requirement.
- ☐ I kept Python as my main DSA and LeetCode language.
- ☐ I did not make *Algorithmic Thinking* mandatory.
- ☐ I did not make CS:APP mandatory.
- ☐ I kept security and reasoning-model work in the default route unless I changed the goal itself.
- ☐ I kept unrelated content-production work outside this repository.
- ☐ I explained every changed prerequisite or book decision.
- ☐ I checked the current edition of every new recommendation.
- ☐ I updated START_HERE.md, CURRICULUM.md, LIBRARY_INVENTORY.md, and README.md together.
- ☐ I recorded the change in Git.

The principle behind the plan

I am not trying to assemble the most academically impressive reading list. I am trying to build the most rigorous route I can sustain, practise, and finish.