DSA

Personal DSA study notes

Algorithm Design

- There will usually be 2 parts in an algorithm

1. remember past states/computations.
2. use that "memory" to perform computation until desired result is obtained.

Notes

1. how to cultivate algorithmic maturity by solving problems?
2. Sliding Window Tips
3. Last Seen Index
4. DFS Guide
5. BFS Guide
6. DP Approaches

Interview Guidelines

1. READ! the problem carefully, DON'T rush to code.
2. Identify ambiguity/questions, try figure them out.
3. Know EXACTLY what inputs are and what desired outputs look like.
4. Play with examples by drawing (visualizing) them first.

- Look for patterns
- Think what data structures are involved (or needs to be involved)

5. Code.

Side Notes

- Limit (pure and rigorous) thinking time to 5 ~ 15 minutes max. Try answering

> "What is the simplest thing I can track to solve this?"

- If after that time you're still debating, choose the **simplest** state variable

Guideline

Thinking mode (≤10 min):

- identify events
- identify state

- identify update rule

Execution mode (immediately after):

- write a rough version
- clean it up later Messy but correct code early beats elegant code late.