



# Above & Beyond CS (ABCS)

## Coding Interview Workshop Series

Workshop 5  
Talking Through Your Solution



# Recap: The Goal of A Coding Interview

...is to get signal on things that we do at Facebook every day.

- How you think about and **tackle hard problems** and how you **communicate** about code
  - Evaluate your problem-solving skills to see if you can translate thought into reasonably correct, well-structured code
- How you consider **engineering tradeoffs** (memory vs. time)
- **Limits** of what you know

# What have we seen so far?



Before coding

...the first 5'

1. Communicate Proactively ✓
2. Design Your Algorithm ✓
3. Work the Clock ✓



During coding

...the next 10'

4. Writing code at the whiteboard ✓
5. Talk through your code / solution
6. Handling mistakes



“After” coding

...the last 2-3'

7. Test your code
8. Ask questions!

# While you're coding...

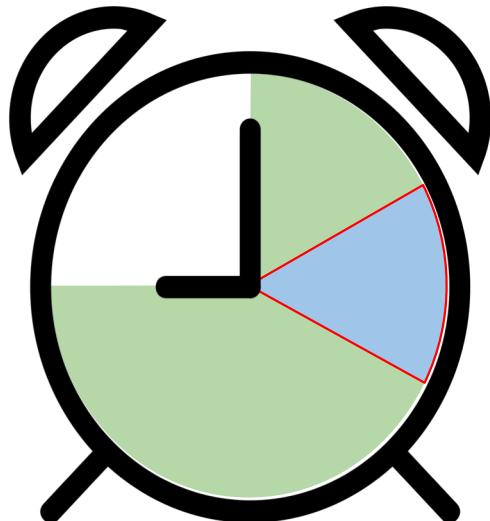
- 4. Write code at the whiteboard
- 5. Talk through your code/solution
- 6. Handling mistakes



## 5. Talk through your code/solution

# Talk Through Your Solution

Keep calm and think out loud.



- Communicate verbally to demonstrate how you tackle hard problems, consider and communicate engineering tradeoffs
- If you aren't talking, an interviewer doesn't know what you're thinking
- This will also help you avoid careless mistakes or catch them when they do happen

# See Interviewers as Collaborators

They want to know what it feels like to work through a problem with you.

- Use “we” instead of “I”
  - e.g. “If we did a breadth-first search we’d get an answer in  $O(n)O(n)$  time.”
- Look up from the whiteboard and talk to the interviewer
  - Make eye contact
  - You might even catch some non-verbal cues about whether or not you’re on the right track



KEEP CALM  
AND  
THINK  
OUT LOUD

# Keep Calm and Think Out Loud

Your interviewer doesn't know what's going on in your head.

- Walk your interviewer through your solution approach and code.
  - Show the interviewer how you're tackling the problem and that you're not stuck
  - As you step through your code, you're more likely to catch any mistakes that you might have made
- Tell them when you're not sure.
  - Let them know what you thought could work, and why it doesn't if you strike out
  - You might even get a hint! 😊

# Ask Questions, Not for Hints

If you're stuck, it's okay to say you don't know.

- Don't try to appear like you know something when you don't
  - Instead say, "I'm not sure, but I'd guess \_\_\_\_ here because \_\_\_\_."
  - The "because" is important – it's an opportunity to rule out other options with poor tradeoffs, or by pulling in other examples from other languages or problems
  - After all, this is an opportunity to demonstrate how you tackle hard problems
- Ask questions, don't ask for hints.
  - Questions aren't just for the first 5 minutes of the interview
  - It's okay to admit you're stumped; ask a question that helps identify the path forward
  - It's never a good idea to ask, "Can you give me a hint?"

# Don't Rush

I know, we told you to work the clock, but rushing will most likely hurt you.

- The time suggestion to finish your code in 10 minutes is a guideline not a rule
- Go ahead at a slow and methodical pace so that you don't commit careless mistakes and so that you can thoroughly think through the problem out loud
- Chances are, you'll finish the problem in less time and with fewer mistakes

# Reminder: Reiterate the Bounds

Don't forget to communicate the space- and time-complexity of your algorithm.

- With your code finished, summarize the big-o of your solution and why it's less than ideal (if necessary)
- Remember that you can:
  - Annotate chunks of your code with their various time and space complexities to demonstrate your understanding of the ode
  - Explain tradeoffs made with regard to time- and space-complexity in your current approach versus alternative approaches



# Thank you!



## Workshop 6

[INSERT REGION DATE/TIME]

### Complete Pre-Work

- ✓ Review screencast
- ✓ Solve HackerRank problems
- ✓ Be prepared to walkthrough your submitted code