

Coding on the Fly

Pitfall 4: Failure to Test Your Solution (Tracing and Edge Cases)



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

By the end of this video you will be able to...

- Recognize the value of testing your draft solution when finishing a technical phone screen

Testing your code

- Let's say you get a draft solution coded.
- Think about what happens if you do this on a computer
 - Compile, error. Run, error.
- You need to do this by playing computer and emulating your code
- In the next video, let's see what Huang does again in the "bad" example

Insert bad Huang video here

- Huang Bad1 3:23->3:53

Yikes!

- So in this video, she doesn't bother to trace through her code and leaves a glaring bug
- She also misses a clear hint from the interviewer, which is another mistake.
- But getting back to testing, let's see what happens in a Huang's "good" example

Insert good Huang video here

- Huang Good2 0:13 -> 1:26

Traced through again

- Nice, by tracing through her code, she's seen a bug. This is exactly why we do this!
- Having a bug or two is okay, but not catching them is really a problem.
- Let's see what she does at the end of the interview again

Insert good Huang video here

- Huang Good2 4:42 -> 5:22

Traced through again

- Great, she double checked it worked!
- The next step would be to consider more corner cases you've been asked to handle
- You might also be prompted by the interviewer to now handle more inputs (with new corner cases)
- So to wrap up – if you're writing your code in a google doc or on a whiteboard, you need to trace code for errors and anticipate corner cases