Why You're Hitting Dead Ends In Whiteboard Interviews

The coding interview is like a maze

Listening vs. holding your train of thought

Finally! After a while of shooting in the dark and frantically fiddling with sample inputs on the whiteboard, you've came up with an algorithm for solving the coding question your interviewer gave you.

Whew. Such a relief to have a clear path forward. To not be flailing anymore.

Now you're cruising, getting ready to code up your solution.

When suddenly, your interviewer throws you a curve ball.

"What if we thought of the problem this way?"

You feel a tension we've all felt during the coding interview:

"Try to listen to what they're saying...but don't lose your train of thought...ugh, I can't do both!"

This is a make-or-break moment in the coding interview. And so many people get it wrong.

Most candidates end up only half understanding what their interviewer is saying. Because they're only half listening. Because they're desperately clinging to their train of thought.

And it's easy to see why. For many of us, completely losing track of what we're doing is one of our *biggest* coding interview fears. So we devote half of our mental energy to clinging to our train of thought.

To understand why that's so wrong, we need to understand the difference between what *we* see during the coding interview and what *our interviewer* sees.

The programming interview maze

Working on a coding interview question is like walking through a giant maze.
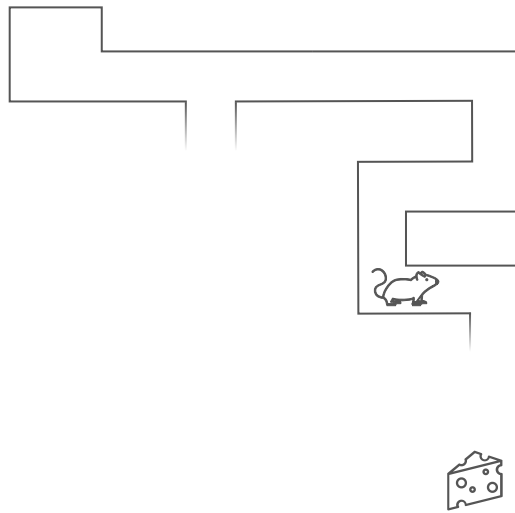
You don't know anything about the shape of the maze until you start wandering around it. You might know vaguely where the solution is, but you don't know how to get there.
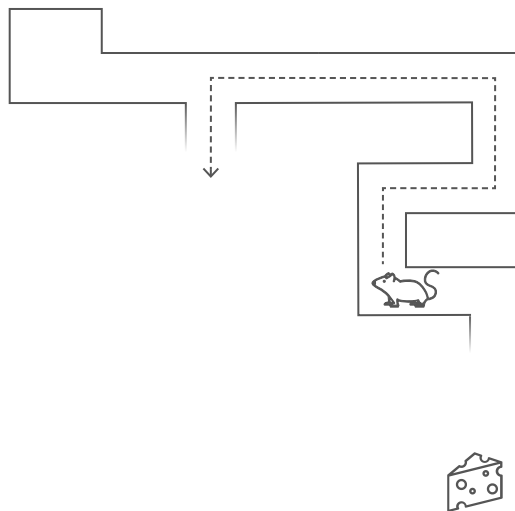

As you wander through the maze, you might find a promising path (an approach, a way to break down the problem). You might follow that path for a bit.
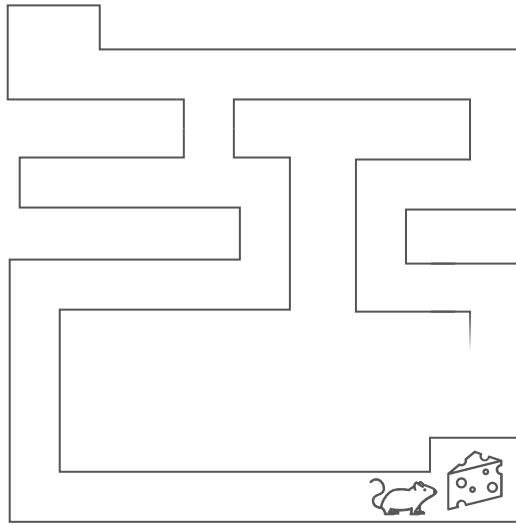
Suddenly, your interviewer suggests a different path:

But from what you can see so far of the maze, your approach has already gotten you halfway there! Losing your place on your current path would mean a huge step backwards. Or so it seems.

That's why people hold onto their train of thought instead of listening to their interviewer. Because from what they can see, it looks like they're getting somewhere!
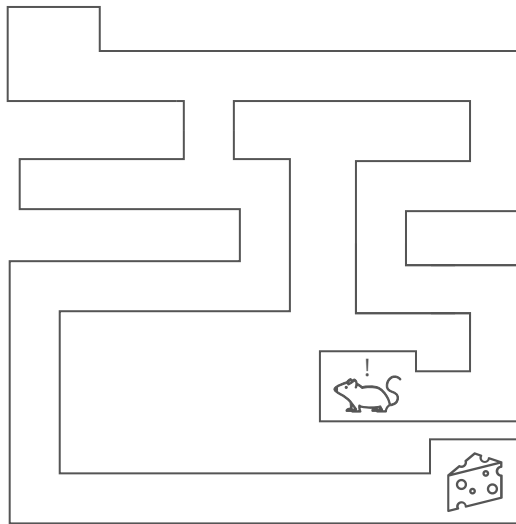
But here's the thing: **your interviewer knows the whole maze**. They've asked this question 100 times.

> I'm not exaggerating: if you interview candidates for a year, you can easily end up asking the same question over 100 times.
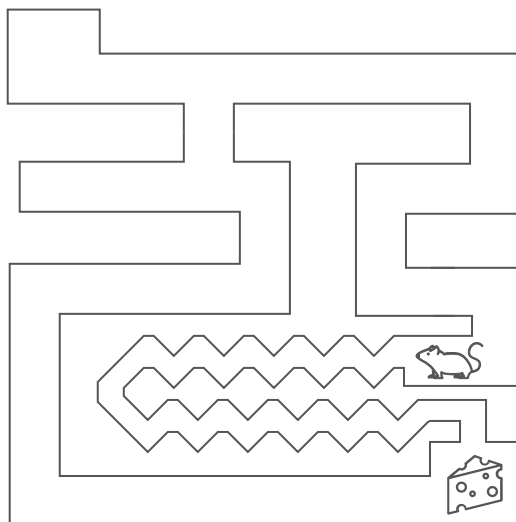
So if your interviewer is suggesting a certain path, you can bet it leads to an answer.

And your seemingly great path? There's probably a dead end just ahead that you haven't seen yet:

Or it could just be a much longer route to a solution than you think it is. That actually happens pretty often—there's an answer there, but it's more complicated than you think.

Hitting a dead end is okay. Failing to listen is not.

Your interviewer probably won't *fault* you for going down the wrong path at first. They've seen really smart engineers do the same thing. They understand it's because you only have a partial view of the maze.

They might have let you go down the wrong path for a bit to see if you could keep your thinking organized without help. But now they want to rush you through the part where you discover the dead end and double back. Not because they don't believe you can manage it yourself. But because they want to make sure you have enough time to finish the question.

But here's something they *will* fault you for: failing to listen to them. Nobody wants to work with an engineer who doesn't listen.

So when you find yourself in that crucial coding interview moment, when you're torn between holding your train of thought and considering the idea your interviewer is suggesting...remember this:

Listening to your interviewer is the *most* important thing.

Take what they're saying and run with it. Think of the next steps that follow from what they're saying.

Even if it means completely leaving behind the path you were on. Trust the route your interviewer is pointing you down.

Because they can see the whole maze.

Want more coding interview help?

Check out **interviewcake.com** for more advice, guides, and practice questions.