Gmail

Siva Kumar K <sivakumar.kumaravelu@gmail.com>

Skip the readings, focus on problems. And use all the hints!

1 message

Parker from Interview Cake <yourfriends@interviewcake.com> Fri, Aug 31, 2018 at 5:00 PM
To: sivakumar.kumaravelu@gmail.com

This is Day 3/7 of our 1-week coding interview email course.

If our problems have felt over your head . . . you'll find this one especially helpful.

But first, really quick: have you looked into our full coding interview prep course yet? (The one that only costs money if it gets you hired.)

It might be the option you've been looking for if you're seeking even more in-depth guidance on interview questions. Now, onto the next tip . . .

The hard part of coding interviews is *getting stuck*—hitting that moment where you don't know what to do next to come up with a solution (or come up with a *more efficient* solution).

The key to getting unstuck—the insight you're missing—is one of those *algorithmic patterns* we talked about. Might be a particular data structure. Might be a way of breaking the problem down.

So to get stuck less often, you should read up on lots of data structures and algorithms, right?

Wrong. Here's the problem:

Just because you're familiar with the data structure or algorithm, doesn't mean you'll know *how to apply it* when a problem calls for it!

Instead, you should be focusing on *building a connection* between the situation that calls for an algorithmic pattern, and the algorithmic pattern itself. The only way to do this is to *run real practice problems*.

That's why on Interview Cake, we *teach* with *problems*. Sure, we have a reading about linked lists ... but we really *teach* the linked list by throwing you into half a dozen questions about them.

Teaching with problems is the only way to really *build those connections*.

But teaching with problems has a side effect: It can feel overwhelming.

Sometimes, folks try a problem or two on Interview Cake, don't get it right away, and conclude that they're not "ready" for our questions. They email me and ask for some pre-readings to get up to speed with data structures and algorithms first.

The problem isn't that they're not ready. It's that they're shy about pressing the "Tell me more" button more than once or twice!

But that button was *made to be pressed*. Heck, press it a couple *dozen* times if you have to. Many of our problems have *that many* hints. And they're there to be used! Use them all. Don't be shy.

When you show all the hints on a question one-by-one, the experience *gracefully degrades* into a *reading*.

You could even say that when you read through a question this way, the question *is* the pre-reading!

But it's more *active* than a normal reading. You have to consider each step on your own before revealing the next one. Much better for learning.

And if you mark the problem for review later, you can try taking a more full swing at it again in a few days.

That is, *if* you buy our full coding interview prep course. Which you could do, if you haven't yet. It only costs money if it works (get the job, or your money back). Hit me up if you have any questions about that.

Later,
Parker
