



rishav ghosh &lt;rishavghosh605@gmail.com&gt;

---

## Why the Leetcode grind doesn't work (and what you can do instead)

1 message

---

**Sam Gavis-Hughson** <sam@byte-by-byte.com>

Mon, Jun 3, 2019 at 9:36 PM

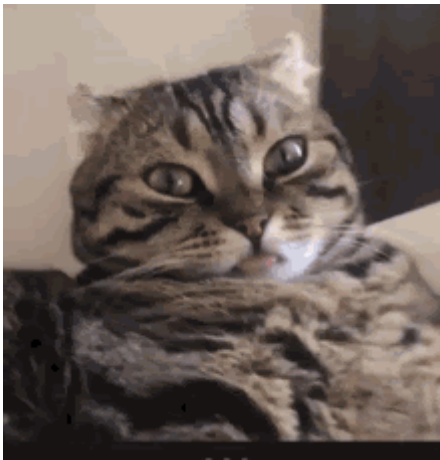
Reply-To: reply-bytebybyte.activehosted.102.504.13780@s4.asa1.acemsd3.com

To: Rishav Ghosh &lt;rishavghosh605@gmail.com&gt;

Hey Rishav -

There's a lot to love about the internet. I mean it gives us cat gifs...

*(Make sure to click "Display images". You know you want to...)*



Dog gifs...



*(Finally I found an excuse to include cute animal gifs in one of my emails!)*

And most of us have jobs because of the internet.

But there's one thing I HATE about the internet...

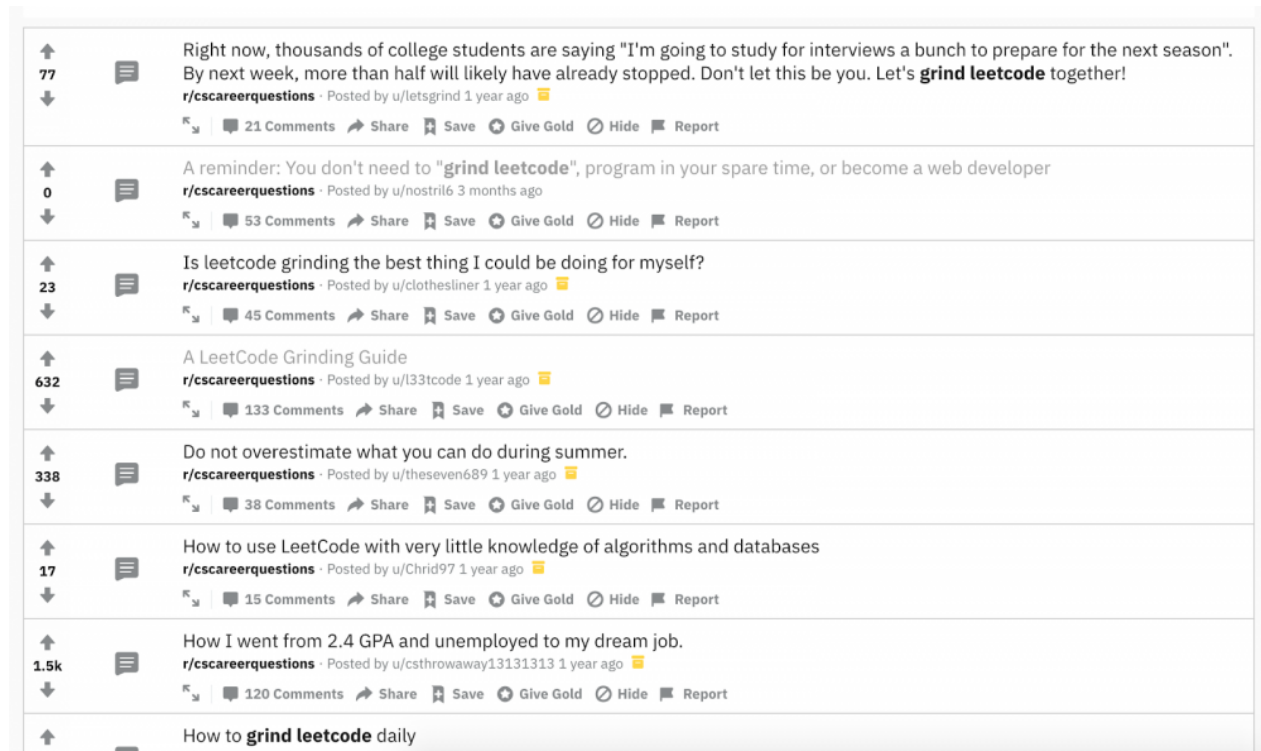
And that is how easy it is for dumb people to spread terrible ideas.

That's how we got **the Leetcode grind**.

In fact, if you search on Reddit, you'll see references to "the grind" coming up again...

And again...

And again...



↑ *This is just a small sampling* ↑

For anyone who is mercifully still in the dark, the Leetcode grind is the idea that the best way to prepare for interviews is simply to go through as many questions as you possibly can.

But there are 3 MAJOR problems with this approach:

## It's demoralizing

You know how many problems there are on Leetcode?

996.

How would you feel if the only way to ace your interview was to learn every single one of those problems?

It's never going to happen...

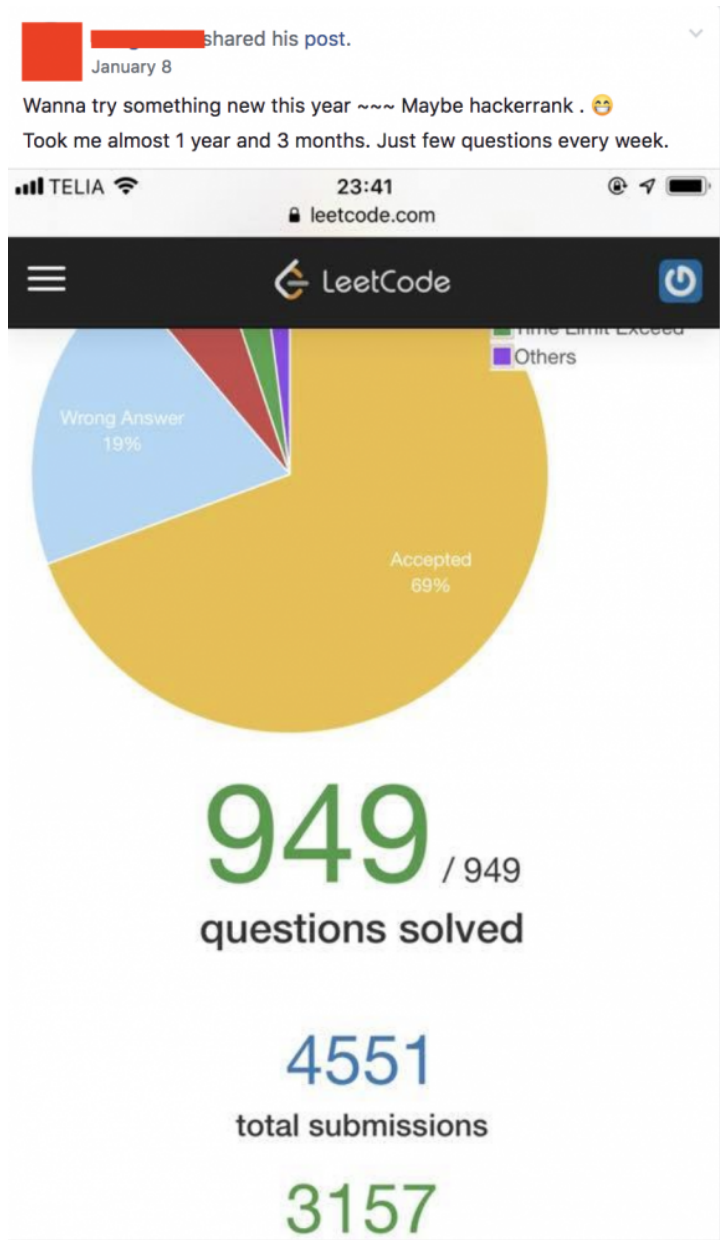
But the grind says you just have to keep doing more. That's the only way to be successful. If you've solved the first 200, they'll ask you problem 201 in your interview. If you've solved 995, they'll ask you problem 996.

We're not going for any strategy here. We're just trying to memorize solutions. But you're always going into the interview with the knowledge that you could have done more.

## It takes a long friggin time

Let's say you decide you are going to solve every problem on Leetcode. How long do you think that's going to take you?

Well for one guy, it took him 1 year and 3 months...



*Notice that Leetcode added a bunch more problems too since he finished this...*

Yeah sure he only did on average 14 questions every week, but do you really want to wait over a year to even INTERVIEW for your dream job?

Or maybe you could do it a little faster. You could solve them all in only 3 months if you do **11 problems every single day...**

You and I both know that ain't gonna happen.

But even if you did...

### It's totally ineffective

If you tried to memorize 996 problems, how many of those would you actually remember at the end of the day?

In one study, they found that when studying languages, passive studiers (AKA people trying to blast through as many problems as possible) had a 34% rate of recall after 1 week.

### That's less than half!

Now imagine what would have happened if they left 2 weeks? Or 2 months? Or God forbid 1 year and 3 months?

But you know what else that study found?

If people used active recall to review and practice what they learned, their rate of recall jumped to 85%. **That's a 250% improvement.**

With our 34%, studying 996 problems means we would only remember 338 of them just ONE WEEK after we had learned them.

**That means  $\frac{2}{3}$  of our time was completely wasted.**

So what does the grind actually mean?

It means that you are putting your interview success at the hands of two 2 key factors that you have zero control over and have a slim chance of actually happening:

1. That your interviewer asks you a question that you have reviewed before
2. That you actually remember the question

Now take the odds of that happening and divide by 5 (you're going to have to repeat this process for multiple different interviewers).

You know what this looks like to me?



↑ *It looks like this crazy guy* ↑

Sure you can do it, but that definitely doesn't make it a good idea.

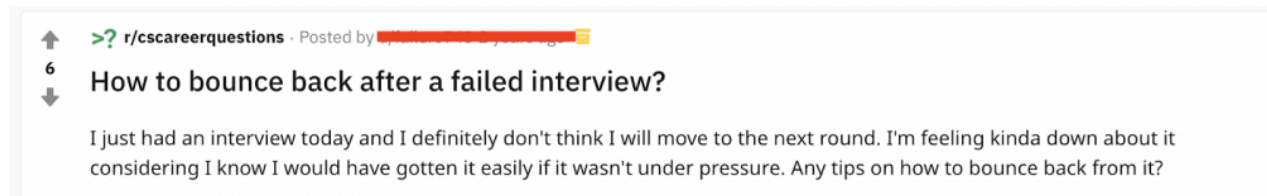
When you prepare by grinding Leetcode, it's ultimately a memory game. The question is simply how many questions can I memorize before my interview.

That leads to so many unfortunate situations. You may have even experienced some of these yourself.

**The "I thought I knew it":**

Have you ever had the experience where you're studying and you think you understand everything but as soon as you're put under pressure everything goes to shit?

That's what happened to this guy and so many others:



Sometimes the pressure really does get to us, but more often than not we didn't actually understand something as well as we thought we did.

As soon as we're put under pressure, we need to understand something twice as well so that our brain doesn't just freak out on us.

### **The "This looks familiar...":**

This is another one where the good ole brain gives us the middle finger.

An alternative title for this might be "coding problem deja vu".

I'm sure you've experienced this one. You see a problem that looks vaguely familiar. Either that or it's similar to a problem you've done before.

So you start trying to follow that thread, but you end up hopelessly stuck.

If you're a talker, you babble incoherently at your interviewer.

If you're a quiet type, you stop saying anything for about 15 minutes.

Either way, the outcome isn't good.

### **The "I've never seen anything like this before":**

The fear of every interviewee... Being asked a question that you've never seen before and having no idea what to do.

If you've been grinding Leetcode, this is basically the end of the road.

You don't have any strategies to get you out of this mess.

You don't have a mental list of common techniques that you can draw on.

You don't even know where to begin.

This is the most embarrassing, painful experience you're likely to experience. It's like me as a freshman interviewing with Microsoft and not knowing what a hashmap was (yup that happened).

And after that, the response to all of these from the grinders? GRIND MOAR.

Spend more time.

Memorize more problems.

Hope and pray that it works.

If you do enough interviews the odds will eventually go in your favor.

Enough! If you want to spend the next 6 months studying, be my guest. If you want to drill question after question day after day until you're finally ready, you can do that.

But for most people, that day will never come. There will never come a day when you feel truly comfortable going into an interview.

You'll delay and delay, postponing judgement day. And then one of 2 things happen. Either you finally get up the courage and go and fail your interview or you keep delaying forever.

But I'd prefer to show you a better way.

Today I want to introduce you to someone who really gets this shit:

Back in 2013, Parker Phinney, had a life-changing experience.



That was the year that his friend, Alice, who was fresh out of a coding bootcamp, came to Parker for help getting a job.

Alice had been putting in the work and was a badass coder, but she really wanted to take her interviewing to the next level.

According to Parker:

*"So we got to work running practice problems, drilling her weak points in algorithms and data structures, and **focusing on how to arrive at the answers, not just what the answers were.***

*"She struggled at first, but it took just one weekend to get Alice to a point where she could crush all the interview questions in my arsenal."*

Shortly after working with Parker, Alice landed a job at Facebook!

After the experience of teaching his Alice, Parker started to hone his own skills. He coached students at multiple bootcamps around San Francisco and began to develop methodologies for helping students to "begin breaking down a new problem and keep moving even when you're stuck".

That was when he realized, like I did early in the history of Byte by Byte, that he truly had a passion for helping students pass their coding interviews.

So he started Interview Cake.

Since 2013, the goal of Interview Cake has been to help as many students as possible pass their coding interviews and get their dream software development jobs.



In that time, they've helped thousands of students get jobs at ever major tech company, including Google, Facebook, Apple, Amazon, and pretty much everywhere else you can think of.

## But what actually is Interview Cake?

Interview Cake is a course designed to help you get exactly what you need to prepare for your coding interviews with none of the fluff.

Parker and his team have built out some of the highest quality material out there and it's been tested rigorously with thousands of students.

Here's how it's different than most other resources out there:

- 50 carefully curated practice problems**  
 Would you rather hundreds of completely random questions in the hopes that you see one of them in your interview or focus on just the most common questions? Interview Cake allows you to get more bang for your buck while actually doing less work because they focus just on 50 of the most common questions rather than showing you every possible edge case.
- Learn the strategies behind the problems**  
 I see so many resources out there that are just Problem + Solution. But you never learn how to bridge the gap between the two. That's the MOST IMPORTANT PART! The team at Interview Cake has written incredibly detailed descriptions not just of the solutions to the problem, but how to think about it so in the future you can come up with the solution on your own.
- 40+ technical articles on every topic you need for success**  
 Still confused after going through a problem? You probably need to revise the underlying fundamentals. Thankfully, Interview Cake has got you covered. Each article highlights exactly what you need to know on a topic and eliminates everything else. Why read a textbook when you can get the Cliffnotes?
- Hints and gotchas for every practice problem**  
 When you're studying on your own, there's usually no one to help you get stuck. The only solution is to look at the answer, which defeats any real learning that you could have had. Interview Cake provides tons of hints for each problem, as well as common gotchas so that you can check for common mistakes without having to just read the solution.

↑ Look at how much detail there is for a single interview question ↑

And that's just some of it. The other thing that I love about Interview Cake is that they have an entire team of people who are continually upgrading and improving the content and overall experience. In the last year, they've added:

- An inline code editor
- [A deep dive into fundamental datastructures](#)
- [A definitive guide to sorting algorithms](#)
- Multiple new practice problems

And the product is continuing to grow.

Obviously, I think Interview Cake is a pretty great course, and I hope you're starting to see why.

In fact, Interview Cake is the product that I turned to when I had an interview back in 2017.

Back in 2017, I was considering whether to stay at my current job, take a new job, or start running Byte by Byte fulltime. As part of that decision, I decided to do a couple of interviews.

At this point, Byte by Byte had already been running for a year and a half, but I felt a little bit like I needed to get out of my own head. Somehow, you get way more clarity when looking at others from the outside than you get when you're in the same situation.

So I turned to Interview Cake. I was sick of having an endless list of practice problems with no end in sight. I knew where to find problems, but I didn't just want to study randomly.

Having a curated list of problems to work through was so valuable to me. I was able to focus my studying specifically on problems that actually mattered and really quiz myself, rather than just jumping straight to the solutions.

I ended up passing multiple rounds of interviews as a result, although I ultimately decided to move to working on Byte by Byte fulltime.

Because I think that Interview Cake is such a valuable resource, I realized that I want to help Parker spread the word, which is part of why I'm sharing this today...



... But I also want to share a very special offer that Parker and I have put together just for Byte by Byte readers.

Parker has agreed to offer a [full 50% off the course price](#) when you sign up through my link.

*(Note that I do receive a commission at no cost to you if you sign up through my link)*

That means you get a full year of access to the entire Interview Cake course for only \$125 (or \$75 for the 3 week quick-study version)!

When you join Interview Cake you can:

- Avoid wasting your valuable study time by focusing specifically on the problems that are most important
- Never get stuck on a brand new problem in your interview by discovering the key problem solving strategies you can use to solve ANY problem



- Master the important points of every key concepts with 40+ detailed technical articles
- Make absolutely certain that you know your stuff by quizzing yourself with the multiple hints and gotchas for each problem

And one of the best parts of all of this is that it's 100% risk-free. I only want you to pay for this course if it's really going to help you and that is one of the things that Parker and I totally agree on.

That's why when you sign up, you also get Interview Cake's **"get the job" guarantee**. If you go through the entire course and are still failing your interviews, they will give you a 100% refund. I love this because you're literally only paying if you get a job offer!

[>> Click here to get Interview Cake for 50% off <<](#)

But because I want to make absolutely sure that you have everything you need to succeed at your interviews, I want to up the ante a little bit.

For everyone who joins Interview Cake through my link, I'm giving away **2 special bonuses for free**.

### **Bonus 1 - Interview and Q&A with Interview Cake Founder Parker Phinney**

Earlier this year, I sat down with Parker for a Q&A and answered a ton of reader questions about preparing for interviews. I gotta say, the value bombs he dropped were pretty crazy.

One of the best ways to learn anything is to study the experts. And Parker is truly an expert. In the call, I interviewed him for about 30 minutes before breaking for Q&A so that we could take about 30 minutes of student questions.

Parker and I agree on a lot of things about the right way to interview for jobs, but we do also have our differences. This recording is a really interesting discussion where you'll get to see both of our perspectives.

### **Bonus 2 - How To Supercharge Your Studying**

For this bonus, I created a brand new live training that I've never taught before. Have you ever had any of the following experiences?

- I keep studying but I don't feel like I'm getting any better
- I don't know what to focus on because there are so many resources
- I just studied X last week and now I feel like I've completely forgotten it

In this training, I teach some concepts that I have never discussed before. We dig deep into interview strategy and how to prepare as effectively as possible.

With the strategies you will learn in this training, you'll be able to get the most out of your Interview Cake subscription and double down on your studying, WITHOUT having to commit any more time.

**I'm giving away both of these bonuses completely free when you sign up!**

I don't recommend many products. I find that most products out there simply don't make the cut. They just aren't that good.

But [Interview Cake](#) is the exception. When I was interviewing for jobs, Interview Cake was where I turned. And I'm excited to be able to share it with you.

[>> I want my FREE bonuses <<](#)

Best,  
Sam

p.s. You don't have to do anything to claim your free bonuses. Just sign up and I'll reach out next week with all the details

Sent to: [rishavghosh605@gmail.com](mailto:rishavghosh605@gmail.com)

6/15/2019

Gmail - Why the Leetcode grind doesn't work (and what you can do instead)

[Unsubscribe](#)

Byte by Byte, [82 Nassau Street, Suite 209, New York, NY 10038, United States](#)