

Coding interview roadmap: How to develop a foolproof interview study plan

Posted by *Sam Gavis-Hughson*



Remember back in the olden days before smartphones? Anyone?

If you wanted to drive somewhere, you had to use this super retro device called a map to figure out where to go.

And you couldn't really just take the map with you and expect to find your way, either. You really had to plan out your route ahead of time to ensure that you got where you were actually trying to go.

Now in this modern age, though, we no longer need maps. With smartphones, you just start driving and your phone figures out where you need to go. It's so simple.

However, as a result, somewhere along the way, we lost the art of planning ahead. These days, I so frequently see people jump into things without any planning at all.

Lack of planning isn't necessarily an issue. In a lot of cases, you can still get where you need to go easily enough. Without a GPS, you'd still eventually figure out how to get where you're going, it just might take longer.

There would be lots of trial and error. You might take one road that leads to a dead end and have to turn around. You might spend an hour driving south when you should have been going north. However, if you drive for long enough, you'll reach your destination.

The same is the case with interviewing for jobs. Most people start preparing without any sort of plan. They just pick something that they think they need to do and start doing it. And sometimes this approach works. If you happen to pick the right things to focus on, you may just get the job.

But oftentimes it doesn't work, and when it doesn't, you can waste a ton of time working on irrelevant tasks. When you're looking for a job, wasting time is probably not something you're particularly keen on doing.

So what if we could plan ahead? What if we had a map to show us the way forwards. It turns out that with a little guidance, it's not that hard for you to develop your own map. In doing so, you will make it possible to get the most out of your limited study time. And in this post, I'm going to show you just how to do that.

Decide on your destination

The first step to planning your interview prep may seem obvious, yet it's something that so many people overlook. You need to know exactly what sorts of jobs you are looking for. In other words, you need to know your destination.

Why is this so important? Because it gives us direction. As Stephen Covey says in the *Seven Habits of Highly Effective People*, "begin with the end in mind".

Knowing what job we want will inform everything about our prep process. It will tell us how we should study, how much time we should spend, and can even help us identify specific skills to cultivate for the prospective job.

For example, let's say you wanted a job at Amazon. If that's the case, there are resources like Glassdoor that will tell you exactly what questions other interviewees were asked. Now you have an idea of exactly what to prepare.

The problem with deciding what jobs to go for, though, is that **most people don't know what they really want**. But they think they do.

The problem with deciding what jobs to go for, though, is that most people don't know what they really want. But they think they do.

[CLICK TO TWEET](#)

When I talk to applicants, especially those still in college, I hear the same few company names over and over again. Google. Facebook. Microsoft. Amazon. Apple.

Occasionally I get a mix of Uber, AirBnb, etc.

Here's the thing, though. What do these companies have in common? Not a whole lot. The one thing that they do have? **Prestige**.

Is that the most important thing in a job? Maybe or maybe not. But getting a job at these companies is going to be a lot of work if the only reason is for the prestige. Google has an acceptance rate of 1%. That means that out of 100 applicants, you need to be better than ALL of them.

Consider the story of John Washam. As he chronicled on the Free Code Camp blog, he spent 8 months practicing FULL TIME for his interview at Google.

And then he didn't get in!

John did end up at Amazon, so clearly it worked out okay for him, but what if he was only interested in Google? **Then he spent 8 months of his life studying his ass off for nothing.**

So you have to ask yourself: "Am I prepared to do a ton of studying to get the job I want when that won't even guarantee me the job?"

Maybe you are, and that's fine, but for the rest of us mere mortals, we need to think a little more deeply about what jobs we are looking for. If you started reading this post thinking "I want to work at Google", that's totally fine. It's always good to have a

starting point.

So how do we figure out what jobs are actually right for us? We will use a technique called the **5 Whys**. Even if you think you know exactly what job you're looking for, I recommend you go through this exercise.

The 5 Whys is a great exercise for developing a deeper understanding of your internal motivations. We can use it to figure out exactly what we're looking for in a job.

The way that it works is simple. To start with, you need to select a job that you are interested in. For argument's sake, let's say Google. Now, you proceed to ask yourself a series of roughly 5 "why"s.

1. **"Why do I want a job at Google?"** Because I want to work with a lot of smart people.
2. **"Why do I want to work with a lot of smart people?"** Because I want to learn as much as I can from others.
3. **"Why do I want to learn as much as I can from others?"** So that I can be a badass developer.
4. **"Why do I want to be a badass developer?"** So that I can get any job I want.
5. **"Why do I want to be able to get any job I want?"** So that I have job security.

The concept behind the 5 Whys is that you keep digging deeper until you ultimately get to your core motivations for something. You don't have to ask exactly 5 "whys", just keep going until you get to the root desire.

In the example above, our root desire is that we want job security. There may be other ancillary reasons, but this is our core motivator.

Once you get to the root desire, you can start considering what are all the options that will meet your root need. For example, if your goal is job security, that could mean finding a job where you get really good at a niche skill. Or it could mean working for a giant company with a pension plan. Or it could mean getting a really high paying job so that you can save up and retire early.

The key is to realize that the company you started with is likely not the only company that can meet your core needs. There may be many companies that fit that criteria.

You also don't necessarily have to have a single core motivation, but you want to figure out what those motivations are and find different jobs that fit those criteria. You will likely come up with a primary motivation and secondary motivations.

For example, your primary motivation could be job security, but secondarily, you also want to have a good work-life balance. You may have to sacrifice some on your secondary motivations to meet your primary motivation, but by understanding your motivations, at least you can choose wisely.

With these motivations in mind, you can come up with different types of jobs that might meet your criteria. Think about both companies and specific types of positions that may be the sort of thing you're looking for.

Once you have several companies that meet your personal goals, it's time to move onto how you will actually get the job.

Set SMART Goals

So we know what sorts of jobs we're looking for, but that's only the first step. It doesn't really help to see the destination if we don't have a route to get there. So let's find that route.

To do this, we are going to use a powerful goal setting framework called SMART goals. SMART stands for Specific, Measurable, Achievable, Relevant, and Time-bounded. Using this framework greatly increases your psychological response to the goal and consequently increases your probability of following through. It also gives us a structured way to assess if we're setting appropriate goals.

Specific

To have a strong goal, it must be specific. Rather than saying that you just want any job, specify what type of job you want or even particular companies. This is why we did the 5 Whys exercise earlier. It helps to clearly see where it is that we're going.

The more specific you can get about your goal, the better. Ideally, your goal should be to get a specific position at a specific company. Or at least a position at one of a list of companies using a specific skill set.

Measurable

We want to know whether we're on track and when we've actually accomplished our goal. If we define a specific enough goal, this should be a given. For example, if our goal is to get a job at Google, it is measurable by considering whether or not we got a job at Google.

Achievable

Achievable simply asks if we can realistically accomplish a given goal. This can be one of the hardest parts of this process to assess on your own because it requires a lot of honest self-reflection. It depends on many factors including how much work you have to do and how much work you are willing to put in. If you're willing to study full time for 8 months (not that I recommend it), what is achievable will be very different than if you only have 30 minutes on alternate Thursdays (also not recommended). This is a very important aspect of smart goals, which I covered in much greater detail here.

Relevant

Chances are you're looking for a job right now, so this is immediately relevant to you, but if a goal isn't something you actually care about, it's unlikely you'll get it done. You'll have no motivation to do it if it's not important to you.

Time-bounded

I don't know about you, but if I don't have a deadline for something, I'll keep putting it off. This is one of the reasons I always recommend you go ahead and schedule your interview. Setting a deadline for your goals is key to your success. It helps you say I need to do exactly this much work every day to get to my goal.

Let's look an example of a goal and how we can make it into a SMART goal.

"I want to get a good job" could become "I want to get a job at Big 4 company making \$150k/year within 6 months."

The first goal is very vague and doesn't help us determine any explicit action steps. The revised goal is:

- **Specific.** We know which companies and the desired salary.
- **Measurable.** It's pretty obvious if we achieve the goal or not.
- **Achievable.** This depends on the individual, but let's assume the goal setter in this case is an experienced software engineer.
- **Relevant.** Again this depends on the individual, but if they're actively searching for a job, it's definitely relevant.
- **Time-bounded.** We have a clear goal of "within 6 months" which tells us when we need to accomplish the goal by.

In the next section, we will see how to use these SMART goals to determine specific daily action steps.

What should I do right now?

So what do we actually do? We have our goal, but it's totally meaningless without specific action.

To do this, we can recursively break down our goals into smaller and smaller pieces. We want to set smaller and smaller SMART goals until we've ultimately broken down all the work that we have to do into 30-minute- to hour-long blocks.

This may sound like a lot of work, but here's the key. We're going to do it in a depth-first search fashion. We're only going to do the specific time blocking for the coming week. For future weeks, we know the big picture of what we're focusing on, so we don't need to get so granular until we get there.

The other big question about this is "How do I know how to break things down?" This is where research comes in. You have to research the sort of job that you're looking for and determine what is going to be required of you so that you can set specific goals.

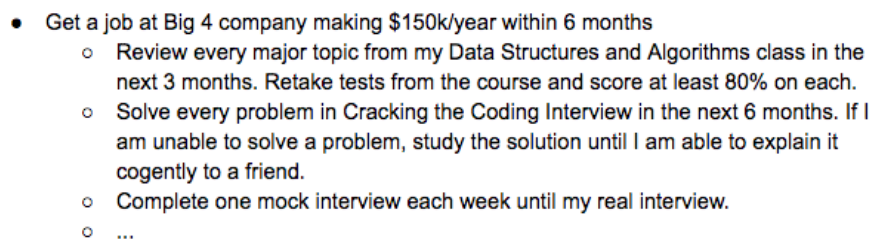
To do this, I recommend doing a close read of the job description, going on Glassdoor, and if possible, talking to someone who has the job that you are looking to get. Each of these will help you gain much greater insights into what the job requirements are and

what to expect from the interview process.

As a high-level starting point, I recommend breaking down your prep into 3 categories: fundamentals, practice problems, and mock interviews. If you struggle to break things down further and need more guidance, you may find it valuable to hire a coach.

Now that we understand the basics, let's put this into practice with an example. I highly recommend using bulleted lists in a Google Doc. It makes it really easy to see the structure of what goals are the parents and children of other goals and I can easily build it out.

To start with, we just lay out the top-level goals. Remember to make them SMART.

- 
- Get a job at Big 4 company making \$150k/year within 6 months
 - Review every major topic from my Data Structures and Algorithms class in the next 3 months. Retake tests from the course and score at least 80% on each.
 - Solve every problem in Cracking the Coding Interview in the next 6 months. If I am unable to solve a problem, study the solution until I am able to explain it cogently to a friend.
 - Complete one mock interview each week until my real interview.
 - ...

From here, we want to continue breaking down the tasks until we know exactly what we need to do this week and today. You'll need to divide up tasks enough so that you can see the overall picture of how you will progress, but you only need to break down the current week on a granular level.

- Get a job at Big 4 company making \$150k/year within 6 months
 - Review every major topic from my Data Structures and Algorithms class in the next 3 months. Retake tests from the course and score at least 80% on each.
 - This week: Strings and Arrays
 - Today: Basic string manipulation
 - Read pages 17 - 19
 - Solve problems 1-5 on page 24
 - ...
 - Next week: Linked Lists
 - ...
 - Solve every problem in Cracking the Coding Interview in the next 6 months. If I am unable to solve a problem, study the solution until I am able to explain it cogently to a friend.
 - This week: Solve the first 7 problems (189 problems/6 months)
 - Today: Solve problem 1
 - ...
 - Next week: Solve the next 7 problems
 - ...
 - Complete one mock interview each week until my real interview.
 - This week: Do a mock interview
 - Today: Find an interviewer and schedule a mock interview
 - Next week: Do a mock interview
 - ...
 - ...

As you can see, I now know exactly what I need to do today:

- Read pages 17 – 19 and solve problems 1-5
- Do the first practice problem in Cracking the Coding Interview
- Schedule a mock interview

By breaking down my studying into this way, I now have precise tasks that I can do. All I have to do is actually follow through and do them.

As you continue through the weeks, you will fill this out more. At the beginning of the week, do some planning. Break down the general tasks for that week into specific actionable chunks.

You may find that things change. That's okay too. Just update the plan as you need to as you go along. I find that too many people stick rigidly to their original goals and this can prevent them from making progress. Don't be afraid of this change.

Do it

The last thing to do is to actually implement your plan. Far too often, I see people set goals or make these elaborate plans, only to have them fall by the wayside. Either that or they spend so much time planning that they leave no time left for implementing.

There are three keys to actually taking action on your plan:

1. **Schedule time**

This is incredibly important. While I have no idea who said it first, you should always remember this quote:

What gets scheduled gets done.

This is critical because if you don't schedule time to study, you will always find other stuff to do. There is always another more important task to do until it is too late. The key is to block off time in your calendar. Make an appointment with yourself to study and do everything in your power to keep that appointment.

2. **Be consistent**

Not only should you schedule study time on your calendar, but I recommend you schedule a manageable period every day. Let's say you want to study 5 hours per week. That is going to be so much easier, and way more productive, if you schedule it in 5 1-hour blocks over the course of the week instead of 1 5-hour block on the weekend. Let's be real, no one can focus for 5 hours straight. I recommend you schedule 30 minutes to 1 hour each day and stick to it. It's not a huge commitment, but it's enough time to see steady progress and reach your goals. And if you try to schedule much more, you risk burnout.

3. **Don't procrastinate**

You're only setting aside limited time, so make sure you get the most of this time. There are many different techniques to mediate procrastination, and I suggest you check out this post on procrastination here.

If you implement these three steps, you'll start making rapid progress. You'll complete different subgoals and be able to go back to your plan and fill it out further.

Every week or so, look back on the past week and see how you did. Did you stick to your goals? Did you make the planned progress? If so, awesome! If not, then maybe it's time to reassess your goals to make them more achievable. It's an iterative process, and it's never going to be right the first time. But as long as you keep improving on it, you'll set yourself up for success.

Having a roadmap for your interview prep makes a huge difference. Instead of wandering around the backroads trying to find your way, you have a clear set of directions. Rather than taking arbitrary turns, you know exactly which turns to take.

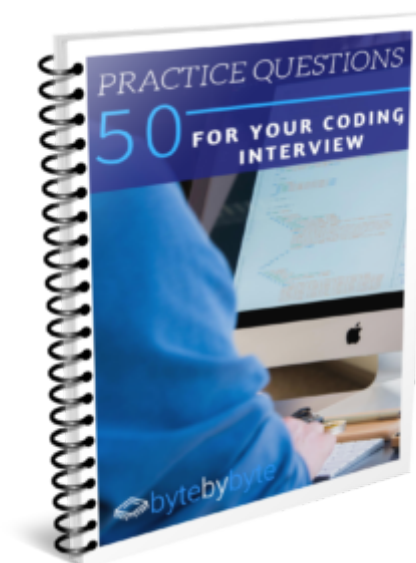
This is a perfect example of the old Abe Lincoln quote about spending the first 4 hours sharpening his axe. This takes time. It's not easy to create this plan. However, if you do it, it will make the rest of your studying so much easier. Skip this step at your own risk.

In the comments below, I'd love to hear your thoughts. What is your one biggest takeaway from this post, and how are you going to implement that right now?

Don't do another coding interview...

...Until you've mastered these 50 questions!

GET YOUR FREE GUIDE





Sam Gavis-Hughson

Sam, founder of Byte by Byte, helps software engineers successfully interview for jobs at top tech companies. Sam has helped thousands of students through his blog and free content -- as well as 400+ paying students -- land jobs at companies such as Google, Amazon, Microsoft, Bloomberg, Uber, and more.

← [Your goals are probably unrealistic...but here's what you can do about it](#)

[Why I turned down Amazon for a company you've never heard of](#) →

4 Comments

Byte By Byte

 Login ▾

 Recommend 5

 Tweet

 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



Suraj Kumar Adhikari • a year ago

Really liked the way you have broken down the steps. I made some plans before but since I didn't put a time limit I ignored it after few weeks of studies. Now I have a framework which I can rely on and move forwards.

^ | v • Reply • Share ›



Sam Gavis-Hughson → Suraj Kumar Adhikari • a year ago

Thanks Suraj! Please update here in a couple weeks and let me know how it's going :)

1 ^ | v • Reply • Share ›



Suraj Kumar Adhikari → Sam Gavis-Hughson • a year ago

Sure I will :)

24 ^ | v • Reply • Share ›



Ergo Proxy • a year ago

Heya Sam just dropping by to let you know that I appreciate the advice that you have been giving out.

Thanks! :)

^ | v • Reply • Share ›



Sam Gavis-Hughson → Ergo Proxy • a year ago

Your welcome! Glad you're finding it useful :)

^ | v • Reply • Share ›



Skanda Bhargav • a year ago • edited

Very well written!

I would like to share a small but very useful tool I use to track where am I spending my time on the computer.

Just a heads-up if your privacy is very dear to you, you may want to stop reading here.

The tool is <https://www.rescuetime.com> and it's helped my track and set realistic goals. I am still on the free plan, and it's been great so far. Having used it close to an year now (started in Mar'17), I can vouch for the increase in productivity. Give it a try and you might end up loving it for giving back you your (precious) time. The Internet can be very distracting and keep us away from reaching our goal.

Note: If links to tools are not allowed, please delete this comment. Also, I'm no way associated with the company that makes this tool. However, I'm extremely satisfied with the tool and thought it might help fellow engineers like us. You can pair all your devices to sum up your total gadget usage.

^ | v • Reply • Share ›



Sam Gavis-Hughson → Skanda Bhargav • a year ago

This is an awesome comment. I totally forgot to mention it in the post, but I love RescueTime personally. Its super nice to be able to track everything automatically. Thanks for sharing!

^ | v • Reply • Share ›



Skanda Bhargav → Sam Gavis-Hughson • a year ago

Thank you!

^ | v • Reply • Share ›



Albert Leng • a year ago

Thank you again Sam!

^ | v • Reply • Share ›



Sam Gavis-Hughson → Albert Leng • a year ago

You're welcome!

^ | v • Reply • Share ›



julian a • a year ago

Great article!

^ | v • Reply • Share ›



Sam Gavis-Hughson → julian a • a year ago

Thanks, Julian!

^ | v • Reply • Share ›



Siddu Siddhartha • a year ago



Awesome !!

^ | v • Reply • Share ›



Sam Gavis-Hughson → Siddu Siddhartha • a year ago

Thanks!

^ | v • Reply • Share ›

ALSO ON BYTE BY BYTE

Coding Interview Question: Merge K Arrays

9 comments • 3 years ago



Sam Gavis-Hughson — You tell me. Which

Coding Interview Question: Build Order

6 comments • 3 years ago



Sam Gavis-Hughson — Awesome!

DYNAMIC PROGRAMMING CRASH COURSE FOR NON-GENIUSES

Download my free guide to learn:

How to finally “get” what Dynamic Programming really is – no Ph.D required

The not-so-obvious way you can solve any dynamic programming problem fast – and not freeze up during your interview

The only 10% of information you need to know to ace your interview – forget all the useless fluff

Enter your email below and get instant access to your free Dynamic Programming guide.

GET THE FREE GUIDE

RECENT POSTS

How To Nail the Amazon Interview: A Practical Guide

The Ultimate Guide to Dynamic Programming

Behavioral Interviews for Software Engineers

Acing the Google Interview: The Ultimate Guide

INTERVIEW CAKE



Interview Cake is an awesome resource for more practice interview questions. Get 50% off for a limited time.

CRACKING THE CODING INTERVIEW



Check out my hands down favorite resource for coding interview prep here.



Made in NYC



sam@byte-by-byte.com



Youtube



© Byte by Byte 2016-2019

[Privacy Policy](#)
[Terms and Conditions](#)

Sam Gavis-Hughson is a software engineer based in New York City. Through Byte by Byte, he publishes regular coding interview question videos, demonstrating proper interview techniques. He also helps many students by offering practice coding interviews to help them get jobs at Google, Facebook, and other exciting tech companies.