

Career Advice

How to get into FAANG - proper plan with sources

A

Anonymous

May 31

Hey,

I am a non-cs software developer working in a small service company - salary is okay (>\$120k). I am an okay software engineer in my company.

I don't have basics in DS or Algo. I know basics in java (between beginner to intermediate) and CMS tool called Adobe Experience Manager (similar to WordPress).

I want to crack FAANG. I know it's a big dream, but I do want to.

What are the resources that can help me? How can I prepare and plan this?

I know about leetcode.

However, apart from practice, I will require the learning of basics about DS, Algo, System design, and anything they may ask.

I need to learn DS, Algo, System Design, Java (almost everything), Any other open source software, etc. and anything else they may ask.

For learning all of these, I will need resources and planning.

I am only interested in online resources. I know there are so many sites - Coursera, Udacity, Udemy, and Youtube. But there are so many. For example, even for DS Algo, Coursera has one course, Udemy has one, Edx has one. SO which one to choose is my confusion, and I only want the best resource.

Suggest the best resources for each topic that Faang interviewer can ask. You can include a few books as well if needed, but I prefer online resources.

I am just looking for the best resource for each topic like DS Algo or Java etc.. If you have two recommendations for some of the topics and you feel that I should go through both, then suggest both.

My major problem is choosing resources as I know lots of resources. So that's why reaching out here.

Also, if you can suggest the planning part that will be helpful.



Questions

1. Suggest the best resources for each topic that Faang interviewer can ask. You may include a few books as well if really needed, but I honestly prefer online resources. Also, I will be happier if you give the one resource recommendations for one topic. If you have two recommendations for some of the topics and you feel that I should go through both, then suggest both.

50 words minimum.

2. My major problem is choosing resources as I know lots of resources. So that's why reaching out here. So only looking for the best resources so I can focus and learn instead of getting confused with multiple resources for a single topic. Also, if you can suggest the planning part that will be helpful.

50 words minimum.

3. Please suggests some level of planning for this if you can. That will help my study. Any other tips or anything will be useful. You can also point to me any online article which can help me in any way.

50 words minimum.

3 Answers

F

FAANG Facebook
Jun 1

+ Ask

1. Suggest the best resources for each topic that Faang interviewer can ask. You may include a few books as well if really needed, but I honestly prefer online resources. Also, I will be happier if you give the one resource recommendations for one topic. If you have two recommendations for some of the topics and you feel that I should go through both, then suggest both.

The resources you need are fairly straightforward. I'm going to make this one long post and copy paste the response because you essentially are asking a derivation of each question in each section of your post.

Instead of simply answering your above post let me give you a rough timeline since you have no prior knowledge.

Before Leetcode get 1 book.

Go through MOST of Cracking the Coding Interview. I say most because there are sections (like the one on math) that are increasingly less important for CS interviews.

Follow the entirety of the book taking copious notes and attempting EVERY problem (NO SHORTCUTS!). You can think of the book as essentially your version of taking college level DS and Algo courses. Once you complete the book and problems you will have benefitted twofold:

1. You will have practiced actual interview problems
2. You will have a good framework for your preparation

3. You will have 90% of the fundamentals down for all possible coding problems
4. You should have / will have identified your weaknesses and know where to focus your study.

Follow the book's guidelines on soft skills, resume prep, study timeline as well. Use it holistically as it was meant, it is an excellent resource.

I know you prefer online resources but this book is the most well rounded comprehensive interview prep book (ie it covers a lot of ground relatively succinctly that you will need for interviews, including system design).

Unfortunately I don't believe this book is always sufficient prep by itself. So after you work through it your next step is to:

1. Get leetcode premium
2. Take the Grokking the System Design course

Use Leetcode Premium specifically to do problems that are tagged to companies you are interested in and to do additional practice on areas you are weak at (ie if you suck at Dynamic Programming sort by all top rated DP problems and do them from easiest to hardest to build your intuition).

Use Grokking the System Design Interview to cover ground not covered in Cracking the Coding Interview (CTCI). CTCI has a lot of great breadth but falls short of depth in some areas, this is one of them.

Your last step / piece of overarching advice here is to create a timeline of when you would like to interview and work backwards. Once you are starting to seriously think about interviewing it is time to both practice problems for time and to do mock interviews (interviewing.io etc). Do your prep only with a pen / paper and a whiteboard (if that is the medium you expect to be interviewing in) and work on time.

Once you hit the interviewing stage it is usually time to start scheduling some interviews you aren't as psyched about for a test run. You need to get used to performing with interview jitters as well.

So to recap:


CTCI (maybe 3 months if no prior knowledge) -> Leetcode + Grokking (a month to start shoring things up) -> Keep Leetcoding / Practice + Doing mocks 1/2 times a week (a month) -> assess your progress if you're ready to start interviewing -> Start doing interviews with lower priority companies -> Schedule interviews with FAANG

2. **My major problem is choosing resources as I know lots of resources. So that's why reaching out here. So only looking for the best resources so I can focus and learn instead of getting confused with multiple resources for a single topic. Also, if you can suggest the planning part that will be helpful.**

The resources you need are fairly straightforward. I'm going to make this one long post and copy paste the response because you essentially are asking a derivation of each question in each section of your post.

Instead of simply answering your above post let me give you a rough timeline since you have no prior knowledge.

Before Leetcode get 1 book.

Go through MOST of Cracking the Coding Interview. I say most because there are sections (like the one c) that are increasingly less important for CS interviews.

Follow the entirety of the book taking copious notes and attempting EVERY problem (NO SHORTCUTS!). You can think of the book as essentially your version of taking college level DS and Algo courses. Once you complete the book and problems you will have benefitted twofold:

1. You will have practiced actual interview problems
2. You will have a good framework for your preparation
3. You will have 90% of the fundamentals down for all possible coding problems
4. You should have / will have identified your weaknesses and know where to focus your study.

Follow the book's guidelines on soft skills, resume prep, study timeline as well. Use it holistically as it was meant, it is an excellent resource.

I know you prefer online resources but this book is the most well rounded comprehensive interview prep book (ie it covers a lot of ground relatively succinctly that you will need for interviews, including system design).

Unfortunately I don't believe this book is always sufficient prep by itself. So after you work through it your next step is to:

1. Get leetcode premium
2. Take the Grokking the System Design course

Use Leetcode Premium specifically to do problems that are tagged to companies you are interested in and to do additional practice on areas you are weak at (ie if you suck at Dynamic Programming sort by all top rated DP problems and do them from easiest to hardest to build your intuition).

Use Grokking the System Design Interview to cover ground not covered in Cracking the Coding Interview (CTCI). CTCI has a lot of great breadth but falls short of depth in some areas, this is one of them.

Your last step / piece of overarching advice here is to create a timeline of when you would like to interview and work backwards. Once you are starting to seriously think about interviewing it is time to both practice problems for time and to do mock interviews (interviewing.io etc). Do your prep only with a pen / paper and a whiteboard (if that is the medium you expect to be interviewing in) and work on time.

Once you hit the interviewing stage it is usually time to start scheduling some interviews you aren't as psyched about for a test run. You need to get used to performing with interview jitters as well.

So to recap:

CTCI (maybe 3 months if no prior knowledge) -> Leetcode + Grokking (a month to start shoring things up) -> Keep Leetcoding / Practice + Doing mocks 1/2 times a week (a month) -> assess your progress if you're ready to start interviewing -> Start doing interviews with lower priority companies -> Schedule interviews with FAANG

- 3. Please suggests some level of planning for this if you can. That will help my study. Any other tips or anything will be useful. You can also point to me any online article which can help me in any way.**

The resources you need are fairly straightforward. I'm going to make this one long post and copy paste the response because you essentially are asking a derivation of each question in each section of your post.

Instead of simply answering your above post let me give you a rough timeline since you have no prior knowledge.

Before Leetcode get 1 book.

Go through MOST of Cracking the Coding Interview. I say most because there are sections (like the one on math) that are increasingly less important for CS interviews.

Follow the entirety of the book taking copious notes and attempting EVERY problem (NO SHORTCUTS!). You can think of the book as essentially your version of taking college level DS and Algo courses. Once you complete the book and problems you will have benefitted twofold:

1. You will have practiced actual interview problems
2. You will have a good framework for your preparation
3. You will have 90% of the fundamentals down for all possible coding problems
4. You should have / will have identified your weaknesses and know where to focus your study.

Follow the book's guidelines on soft skills, resume prep, study timeline as well. Use it holistically as it was meant, it is an excellent resource.

I know you prefer online resources but this book is the most well rounded comprehensive interview prep book (ie it covers a lot of ground relatively succinctly that you will need for interviews, including system design).

Unfortunately I don't believe this book is always sufficient prep by itself. So after you work through it your next step is to:

1. Get leetcode premium
2. Take the Grokking the System Design course

Use Leetcode Premium specifically to do problems that are tagged to companies you are interested in and to do additional practice on areas you are weak at (ie if you suck at Dynamic Programming sort by all top rated DP problems and do them from easiest to hardest to build your intuition).

Use Grokking the System Design Interview to cover ground not covered in Cracking the Coding Interview (CTCI). CTCI has a lot of great breadth but falls short of depth in some areas, this is one of them.

Your last step / piece of overarching advice here is to create a timeline of when you would like to interview and work backwards. Once you are starting to seriously think about interviewing it is time to both practice problems for time and to do mock interviews (interviewing.io etc). Do your prep only with a pen / paper and a whiteboard (if that is the medium you expect to be interviewing in) and work on time.

Once you hit the interviewing stage it is usually time to start scheduling some interviews you aren't as psyched about for a test run. You need to get used to performing with interview jitters as well.

So to recap:

CTCI (maybe 3 months if no prior knowledge) -> Leetcode + Grokking (a month to start shoring things up) -> Keep Leetcoding / Practice + Doing mocks 1/2 times a week (a month) -> assess your progress if you're ready to start interviewing -> Start doing interviews with lower priority companies -> Schedule interviews with FAANG



1. Suggest the best resources for each topic that Faang interviewer can ask. You may include a few books as well if really needed, but I honestly prefer online resources. Also, I will be happier if you give the one resource recommendations for one topic. If you have two recommendations for some of the topics and you feel that I should go through both, then suggest both.

So personally I don't look at videos, those are a waste of time to me.

First of all, when you learn about data structure and algorithms, you need to choose a language to start with. Of course, these ideas exist in most language, but there are certainly languages with better support data structure (for instance, C barely has any built-in data structure).

2. My major problem is choosing resources as I know lots of resources. So that's why reaching out here. So only looking for the best resources so I can focus and learn instead of getting confused with multiple resources for a single topic. Also, if you can suggest the planning part that will be helpful.

Since you wrote Java, I would suggest start learning things around Java.

A good website I use is www.geeksforgeeks.org

For Java, use this: <https://www.geeksforgeeks.org/java/>

Data structures: <https://www.geeksforgeeks.org/data-structures/>

Algorithms: <https://www.geeksforgeeks.org/fundamentals-of-algorithms/#AnalysisofAlgorithms>

You can do it in whatever order you preferred. I would start with learning Java first, then into data structures and algorithms.

3. Please suggests some level of planning for this if you can. That will help my study. Any other tips or anything will be useful. You can also point to me any online article which can help me in any way.

For System design, I would learned some design patterns (<https://www.geeksforgeeks.org/software-design-patterns/>), then just go straight to leetcode question.

System design is something you learn as you write more code, if you aren't able to learn it in your job, just do questions on it, that's faster for the sake of interview.

Also, if you are interviewing for entry level job, they probably won't test you on System Design anyways.

Reading articles is a lot faster than watching videos (to me at least), so I would start with reading articles in [geeksofgeeks](http://www.geeksforgeeks.org), and if there's something I really can't understand, I might search youtube or Stack Overflow (<http://stackoverflow.com>)

Google is also your best friend.

Lastly, for the sake of interview, leetcode is still way more important than learning DS and algorithm. Of course, in order to perform well at FAANG you definitely need solid foundation.

A Anonymous ★★★★★

days answered all my questions patiently. I was really happy.

1. Suggest the best resources for each topic that Faang interviewer can ask. You may include a few books as well if really needed, but I honestly prefer online resources. Also, I will be happier if you give the one resource recommendations for one topic. If you have two recommendations for some of the topics and you feel that I should go through both, then suggest both.

I used to have the exact same problem as you -- not knowing where to look because there are so many resources out there. I think it's a good approach to choose one or two resources, make sure that they're great, and just be consistent with them.

With that said, for data structures/algorithms, I'd recommend Geek for Geeks (link below). They're the best resource that I've come across because they go over exactly what you need to know AND they provide the implementation. The best thing you can do for yourself with prepping for data structures/algorithms is to be absolutely sure that you know how to implement it from scratch. Once you have this down, I'd recommend sticking to leetcode and getting in as many easy/medium questions as possible.

<https://www.geeksforgeeks.org/data-structures/>

For system design, I've had a lot of success with Grokking the System Design curriculum. Below is a github repo outlining it. Would highly recommend going through the examples and purchasing the course if you need more guidance.

<https://github.com/tuliren/grokking-system-design>

2. My major problem is choosing resources as I know lots of resources. So that's why reaching out here. So only looking for the best resources so I can focus and learn instead of getting confused with multiple resources for a single topic. Also, if you can suggest the planning part that will be helpful.

As I mentioned in the answer above, the important part is to choose one or two resources and stick to them.

Data Structures Learning: Geek for Geeks

Data Structures Practice: Leetcode (Try to do >100 easy/medium questions)

System Design: Grokking the coding interview. Go through the examples and purchase the course if you need more guidance.

3. Please suggests some level of planning for this if you can. That will help my study. Any other tips or anything will be useful. You can also point to me any online article which can help me in any way.

What I've found helpful in landing offers from top tech companies is to be consistent. I'd suggest doing 30-60 minutes each day during the week, and 2-3 hours each day on the weekend.

Once you have a good foundation, you can then focus on doing problems and rotate through each topic. I'd recommend reading this article about how a bootcamp grad got a job at airbnb that paid \$250,000 and what his interview prep process was like. Good luck!!

<https://haseebq.com/how-to-break-into-tech-job-hunting-and-interviews/>



Find the answers you need by posting a project

[+ Ask Questions](#)







