Gmail

Neeraj Dhakad <neerajbamboriya2@gmail.com>

Day 1: DSA, baby!

InstaByte <newsletter@mail.interviewmaster.io>
Reply-To: InstaByte <newsletter@mail.interviewmaster.io>
To: "neerajbamboriya2@gmail.com" <neerajbamboriya2@gmail.com>

Thu, Feb 6, 2025 at 11:05 PM

INTRODUCTION

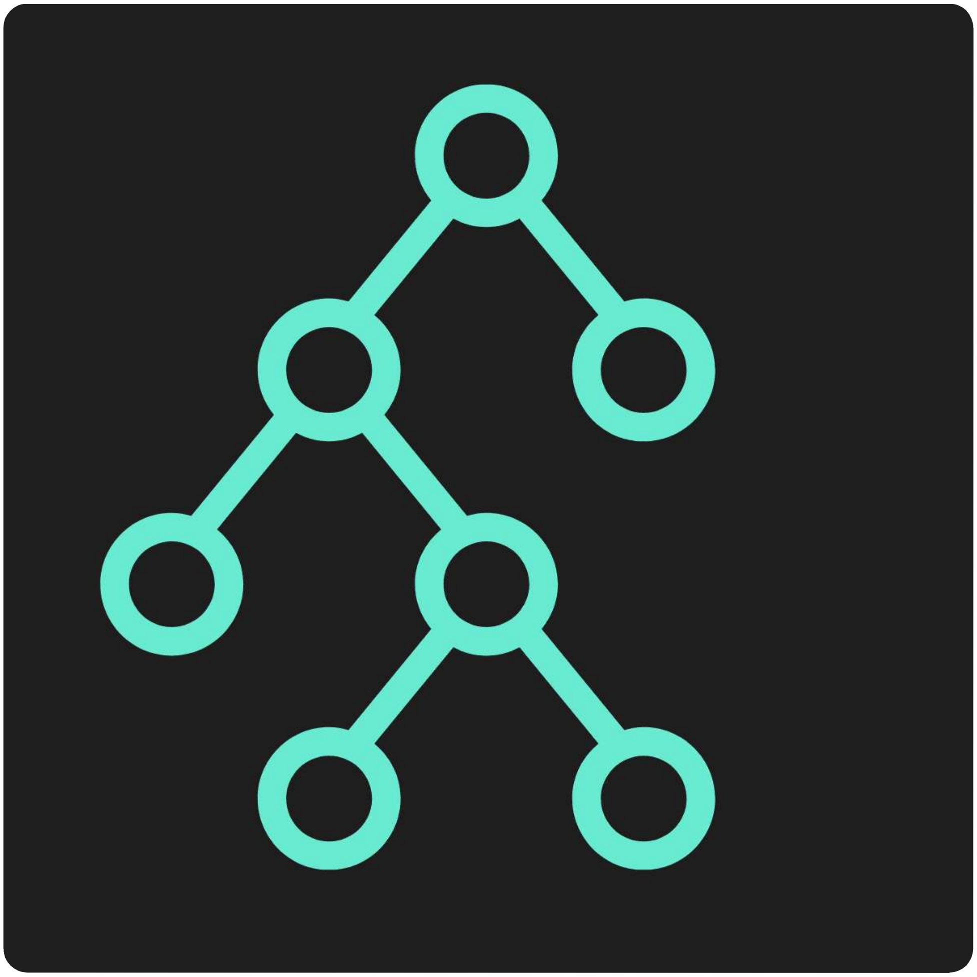
Hi there!

Welcome to Day 1 of Interview Master! Are you pumped to start this journey with me? Let's dive right in!

I took 127 interviews before I cracked Google and Amazon. More than 90% of the problems that I was asked come from this one course. This course is called Data Structures and Algorithms (DSA). Today, we’ll learn how to master DSA.

Read time: under 4 minutes

DECODING DSA



What are Data Structures and Algorithms?

Remember the good old days when you used physical dictionaries to look up the meaning of a word. Yes, I’m talking about those chunky books with a zillion pages. Let’s say you need to find the word “programming” in a dictionary, how would you do it?

You would open the dictionary. Let’s say you landed on a page with words starting with “m”. You know that “m” comes before the “p” of “programming”. The word “programming” must come after the page you are on. So, you start looking for “programming” in the right half of the dictionary. You’ll repeat this step multiple times until you find the word. The process or in other words, the “Algorithm” that you used is called “Binary Search". And the way data is stored, which is in a dictionary, is called a “Data Structure”.

How to Master DSA?

I used Princeton University’s Algorithms 1 and Algorithms 2 to learn DSA. Prof. Sedgewick, who is one of the instructors of this course, has been teaching Algorithms since the 1980s. He has even invented a few algorithms himself. His book on Algorithms has been a bestseller for decades. Needless to say, he knows what he’s talking about.

The Code Snippets used in this course are in Java. If you don’t know Java, you can use ChatGPT to translate code into any other language. But if you don’t want to go through the hassle and Java is deal breaker for you, you can check out these other courses:

- IBM: Data Structures & Algorithms Using C++
- Google’s Data Structures and Algorithms in Python

3 MISTAKES TO AVOID

Here are 3 mistakes I made when I was learning DSA:

1. **Learn “everything”**: There are hundreds of algorithms out there. And many of them are almost never asked in the interviews. Don’t try to learn everything.
2. **Focus on code**: I was a beginner in programming when I was learning DSA. And the code implementation of algorithms looked very scary at the time. Don’t worry too much about the code if you’re having trouble. Focus on building your logic.
3. **Too much time on theory**: I spent too much time learning the theory. But real learning happens when you apply these algorithms to actual problems. I’ll show you how to do this tomorrow.

HIDDEN TREASURE

Congratulations! You’ve found the first Hidden Treasure. When it comes to DSA, it’s very hard to know how much is enough? If you want to know the topics you need to cover before you can move onto the next step, you can unlock the treasure here:

Refer to Unlock

📌 Or copy and paste your referral link to others: https://instabyte.io/subscribe?ref=A6lbQCYndC

Share your referral link on LinkedIn or with your entire class to unlock the treasures quicker.

WHAT’S NEXT

So far, we have only covered the theory you need to know to prepare for interviews. Next, I will give you practical tips on how to use this theory to ace your interviews.

If you’re liking this course, you can follow me on Linkedin for more interview tips and tricks.

See you tomorrow!

Cheers,
Sahil

228 Park Ave S, #29976, New York, New York 10003, United States

Gmail **Neeraj Dhakad <neerajbamboriya2@gmail.com>**

Day 2: Leetcode Cheatcode

InstaByte <newsletter@mail.interviewmaster.io> Fri, Feb 7, 2025 at 9:01 AM
Reply-To: InstaByte <newsletter@mail.interviewmaster.io>
To: "neerajbamboriya2@gmail.com" <neerajbamboriya2@gmail.com>

INTRODUCTION

I’ve missed you. Welcome back!

Yesterday was all about building a theoretical foundation to crack coding interviews. Today, we will focus on applying our knowledge to solve real interview problems. We’ll do this with the help of a very popular platform called Leetcode. I’ll teach you everything I learned by solving 541 Leetcode problems.

Read time: under 4 minutes

LEETCODE CHEATCODE

Why Leetcode?

There are 3 main reasons to use Leetcode:

1. **Practice the right problems:** LeetCode has tons of questions like those used in real coding interviews. So you're practicing the exact skills companies ask for. It's like having a cheat sheet for the test!
2. **Level up your coding skills:** LeetCode questions help you think creatively and solve problems efficiently. They will improve your logic, problem-solving, and coding speed.
3. **Learn from others:** LeetCode has a huge community of programmers ready to help. You can read discussions, ask questions, and even see different approaches to solving problems.

Controversial Opinion

Majority of the interview rejections that I faced came in the initial days when I was using Java in interviews. Java is verbose and has a lot of boilerplate code. As a beginner, I used to get lost in the angle brackets and semicolons. This syntax overhead made it very hard to focus on the most important thing, which is “The Algorithm”.

Somewhere around my 100th interview, I decided to switch to Python. And it was a complete game changer. My speed of thinking as well as coding increased and I started solving 2-3 problems within the same interview. Many of my friends who switched to Python have seen similar results. If you can, learn Python even if it's just for the coding interviews.

Where to start?

I highly recommend that you start with the Top Interview 150 on Leetcode. This list has the Top 150 most asked coding problems. Sort these problems by difficulty and go from easy to medium to hard.

DOs and DON'Ts

❌ Don't spend more than 60 minutes on a problem. Learn from the most upvoted solutions after 60 minutes and move on.
❌ Don't try Leetcode mediums in the beginning. Start with easy problems and slowly progress towards medium problems.
❌ Don't worry about Leetcode hards. Most problems asked in the interviews are medium difficulty.
❌ Don't try to optimize your code for total lines of code. Optimize for readability (and big-O).

✅ Time yourself. If you can't solve a problem in less than 45 minutes, you will not be able to solve it in an interview.
✅ Always read the top 3 most upvoted solutions. Even if you manage to solve a problem, there's a good chance someone else has solved it better than you.
✅ Don't compare yourself to others. Everyone starts from zero. I could not solve my first Leetcode problem for 8 hours. Keep going!

HIDDEN TREASURE

Congratulations! You've found a Hidden Treasure. In the 100+ interviews that I gave, I have seen 5 algorithms repeat again and again. If you want to know what these algorithms are, unlock the treasure here:

Refer to Unlock

📌 Or copy and paste your referral link to others: https://instabyte.io/subscribe?ref=A6lbQCYndC

Share your referral link on LinkedIn or with your entire class to unlock the treasures quicker.

WHAT'S NEXT

Now that you're ready to apply, you need to prepare a killer resume that the recruiters will love. On Day 3, I'll share the resume that got me into Google.

You're on a roll! You've completed Day 2. If you are finding value in this course, you can follow me on Linkedin for more interview tips and tricks.

Until next time!

Cheers,
Sahil

Update your email preferences or unsubscribe here

228 Park Ave S, #29976, New York, New York 10003, United States

Sahil **Gaba**

☎XXX-XXX-XXX | ✉ xxx@gmail.com | in gabag26

SKILLS

BACK END DEVELOPMENT| Java • Python • Spring • Express • NodeJS
DATA PIPELINES| Amazon Redshift • Amazon EFS • S3
MISCELLANEOUS| Amazon AWS • Recommendations • Collaborative filtering • Machine learning • MongoDB
FRONT END DEVELOPMENT| ReactJS • Redux • JavaScript • Bootstrap • Materialize • HTML • CSS
SOFT SKILLS| Team player • Bias for action • Deliver results

EXPERIENCE

AMAZON | **Java, Spring, Python, AWS, Machine Learning, Recommendations**
Software Engineer | Jul 2019 – Present, Seattle, WA

Amazon's Choice recommendations for incomplete missions

- Ideated and developed a new strategy to recommend Amazon's Choice items related to customer's incomplete missions
- Led technical architecture discussions with Amazon's Choice and Amazon API teams and designed user experience with Amazon Gateway team
- Built a data pipeline with Amazon Redshift and Amazon EFS to use offline data, thereby limiting traffic for Amazon's Choice service and eliminating scalability issues

Personalized recommendations with Topic Modeling

- Implemented a recommendation feature in Java using topics and incomplete missions of customers to improve recommendations on Amazon
- Built an extensible 'filters' module to remove Adult topics from final dataset

Complementary recommendations for Hardlines

- Ideated and developed a new algorithm using Collaborative Filtering to improve coverage of complementary recommendations for 500 million products on Amazon
- Created a data pipeline using Amazon EFS, Amazon Distributed Data Service and Amazon Distributed Job Service to create fresh dataset at regular intervals
- Implemented multiprocessing and LRU cache in Python to solve scalability challenges
- Conducted A/B tests and used Bayesian inference to launch the feature on Amazon

Mitigating the impact of Black Hole Effect on Recommendation Service

- Performed stress test on the service after artificially creating black hole effect. Black Hole Effect occurs when unhealthy host gets disproportionate share of requests because Load Balancer is configured to use least connections algorithm
- Alleviated the impact of black hole effect by adding time delay to error responses

Experiments altogether generated more than $X million incremental revenue for Amazon and played a pivotal role in achieving annual team goal

FINTECH CORPORATION
Software Engineer | Jan 2017 – Jun 2019, Chicago, IL

- Implemented Fidessa API for custom order management and execution
- Developed and tested regional solutions tailored for stock exchanges of Americas

EDUCATION

UNIVERSITY OF ILLINOIS – URBANA CHAMPAIGN (UIUC)
M.S., Mechanical Engineering
Dec 2016 | Urbana-Champaign, IL
GPA: 3.74/4

INDIAN INSTITUTE OF TECHNOLOGY DELHI (IITD)
B.S., Mechanical Engineering
May 2014 | New Delhi, India
GPA: 8.98/10

HONORS & AWARDS

CHARPAK SCHOLARSHIP | For top 35 exchange students from India| French Embassy | 2012 | New Delhi

DIRECTORS MERIT AWARD | For top 7 percent students in the class | IIT Delhi | 2011 | New Delhi

K. VASUDEVAN AWARD | For topping the institute among 850 students| IIT Delhi | 2011 | New Delhi

COURSEWORK

Data Structures and Algorithms
Developing Android Apps
Machine Learning
Data Mining
Applied Statistics
Linear Algebra
Differential Equations
Robotics: Mechanical Design

INTERESTS

Traveling • Fitness • Nutrition • Food • Self-improvement

Gmail

Neeraj Dhakad <neerajbamboriya2@gmail.com>

Day 3: My Google Resume

InstaByte <newsletter@mail.interviewmaster.io> Fri, Feb 7, 2025 at 8:31 PM
Reply-To: InstaByte <newsletter@mail.interviewmaster.io>
To: "neerajbamboriya2@gmail.com" <neerajbamboriya2@gmail.com>

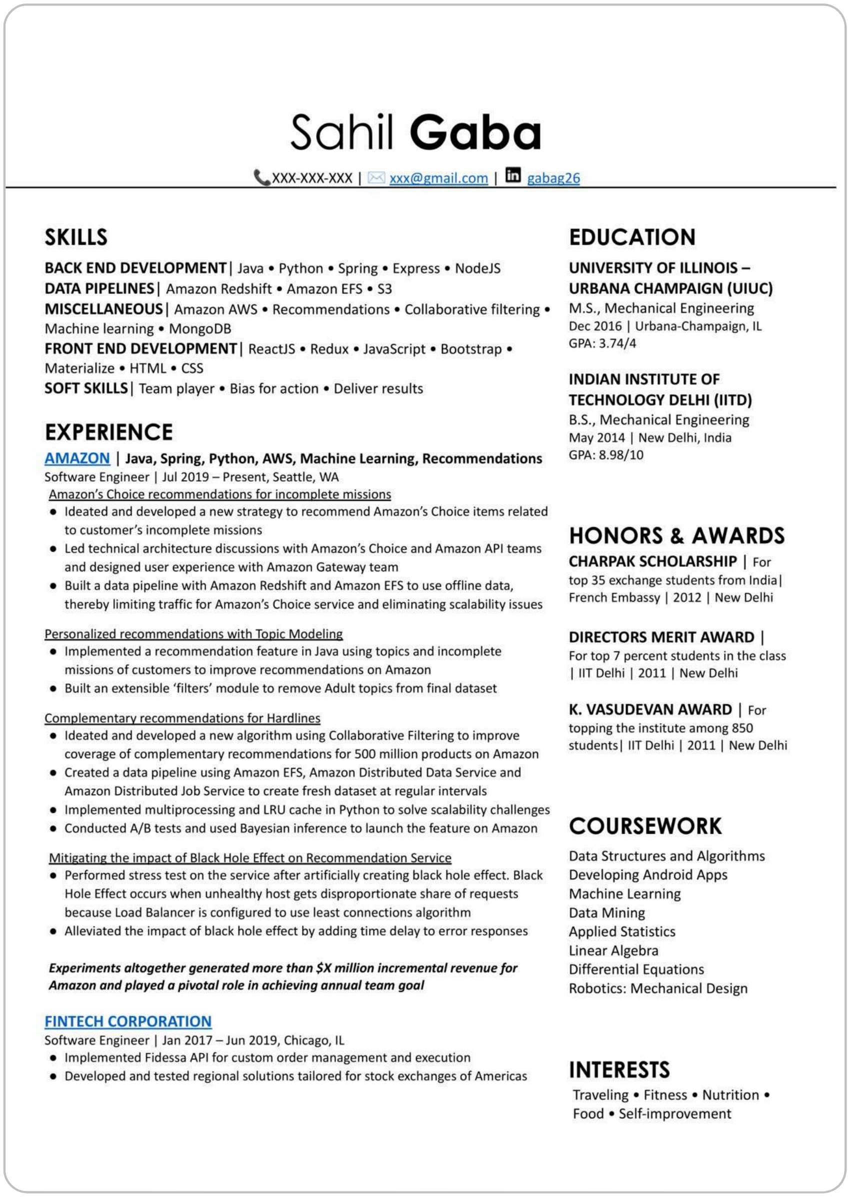
INTRODUCTION

Welcome to our third date. We won't just be acquaintances anymore.

Today I’m going to share something that I usually don’t share until the fifth date. I’ll walk you through the resume that got me into Google. Let’s make sure that when you hit that apply button, you don’t hear crickets.

Read time: under 3 minutes

MY GOOGLE RESUME



Sahil **Gaba**

📞XXX-XXX-XXX | ✉ xxx@gmail.com | in gabag26

SKILLS

BACK END DEVELOPMENT| Java • Python • Spring • Express • NodeJS
DATA PIPELINES| Amazon Redshift • Amazon EFS • S3
MISCELLANEOUS| Amazon AWS • Recommendations • Collaborative filtering • Machine learning • MongoDB
FRONT END DEVELOPMENT| ReactJS • Redux • JavaScript • Bootstrap • Materialize • HTML • CSS
SOFT SKILLS| Team player • Bias for action • Deliver results

EXPERIENCE

AMAZON | Java, Spring, Python, AWS, Machine Learning, Recommendations
Software Engineer | Jul 2019 – Present, Seattle, WA

Amazon's Choice recommendations for incomplete missions

- Ideated and developed a new strategy to recommend Amazon's Choice items related to customer's incomplete missions
- Led technical architecture discussions with Amazon's Choice and Amazon API teams and designed user experience with Amazon Gateway team
- Built a data pipeline with Amazon Redshift and Amazon EFS to use offline data, thereby limiting traffic for Amazon's Choice service and eliminating scalability issues

Personalized recommendations with Topic Modeling

- Implemented a recommendation feature in Java using topics and incomplete missions of customers to improve recommendations on Amazon
- Built an extensible 'filters' module to remove Adult topics from final dataset

Complementary recommendations for Hardlines

- Ideated and developed a new algorithm using Collaborative Filtering to improve coverage of complementary recommendations for 500 million products on Amazon
- Created a data pipeline using Amazon EFS, Amazon Distributed Data Service and Amazon Distributed Job Service to create fresh dataset at regular intervals
- Implemented multiprocessing and LRU cache in Python to solve scalability challenges
- Conducted A/B tests and used Bayesian inference to launch the feature on Amazon

Mitigating the impact of Black Hole Effect on Recommendation Service

- Performed stress test on the service after artificially creating black hole effect. Black Hole Effect occurs when unhealthy host gets disproportionate share of requests because Load Balancer is configured to use least connections algorithm
- Alleviated the impact of black hole effect by adding time delay to error responses

Experiments altogether generated more than $X million incremental revenue for Amazon and played a pivotal role in achieving annual team goal

FINTECH CORPORATION
Software Engineer | Jan 2017 – Jun 2019, Chicago, IL

- Implemented Fidessa API for custom order management and execution
- Developed and tested regional solutions tailored for stock exchanges of Americas

EDUCATION

UNIVERSITY OF ILLINOIS – URBANA CHAMPAIGN (UIUC)
M.S., Mechanical Engineering
Dec 2016 | Urbana-Champaign, IL
GPA: 3.74/4

INDIAN INSTITUTE OF TECHNOLOGY DELHI (IITD)
B.S., Mechanical Engineering
May 2014 | New Delhi, India
GPA: 8.98/10

HONORS & AWARDS

CHARPAK SCHOLARSHIP | For top 35 exchange students from India| French Embassy | 2012 | New Delhi

DIRECTORS MERIT AWARD | For top 7 percent students in the class | IIT Delhi | 2011 | New Delhi

K. VASUDEVAN AWARD | For topping the institute among 850 students| IIT Delhi | 2011 | New Delhi

COURSEWORK

Data Structures and Algorithms
Developing Android Apps
Machine Learning
Data Mining
Applied Statistics
Linear Algebra
Differential Equations
Robotics: Mechanical Design

INTERESTS

Traveling • Fitness • Nutrition • Food • Self-improvement

At the top, I have my name in large font and my contact details. Fairly standard.

Below it on the left, I have summarized all my skills in one place. Most companies have these automatic filters called ATS filters. These filters might be rejecting you based on required skills. You want to make sure that all your skills are listed on the resume. Having said that, you still need to clear the interview. Only include the skills that you would not mind getting tested on.

Next I have my work experience. If you look closely, I have tried to answer three questions for every project that I've mentioned

1. What did I do?
2. How did I do it?
3. What was the impact?

Project title and the first line give the context which is the "what" part. The next couple of lines go on to explain the steps I took to accomplish it, which is the "how" part. At last, we have one line explaining the impact. This line would usually contain some numbers to quantify the impact.

Let's move on to the right section. At the top, I have education. In the context of software engineering jobs, your education and gpa are less important than your actual skills. So, I've kept it in the right section. Below it, I have some awards I received, coursework and my hobbies. Given that I have some industry experience now, I don't think these things are that important anymore. So, I might remove them when I apply for the next job.

HIDDEN TREASURE

You've found yet another Hidden Treasure. If you want to use the editable version of my resume, you can unlock the treasure here:

Refer to Unlock

📌 Or copy and paste your referral link to others: https://instabyte.io/subscribe?ref=A6lbQCYndC

Share your referral link on LinkedIn or with your entire class to unlock the treasures quicker.

3 MISTAKES TO AVOID

1. **Content over template:** Though I have shared my resume template with you today, I would like you to know that the content of the resume is much more important than the template. So don't pay for expensive resume templates.
2. **Don't rate yourself:** In my college days, I used to think that I am the best looking guy but the girls in my college disagreed. Don't rate yourself on your own skills.
3. **Always update:** What was relevant in your last job search might not be relevant anymore. For example, coursework and old awards etc. are not relevant anymore for my resume. Don't just copy and paste your last resume.

WHAT'S NEXT

Congratulations! You've made it to the interview. I'll share the interview tips on Day 4.

Day 3 is done. You're killing it! If you are finding this course useful, you can follow me on Linkedin for more interview tips and tricks.

I'll see you tomorrow!

Cheers,
Sahil

228 Park Ave S, #29976, New York, New York 10003, United States

Gmail

Neeraj Dhakad <neerajbamboriya2@gmail.com>

Day 4: Don’t miss the interview day

1 message

InstaByte <newsletter@mail.interviewmaster.io> Sat, Feb 8, 2025 at 9:06 PM
Reply-To: InstaByte <newsletter@mail.interviewmaster.io>
To: "neerajbamboriya2@gmail.com" <neerajbamboriya2@gmail.com>

INTRODUCTION

We’ve come so far. We don’t want to mess up now.

Welcome to Day 4 of Interview Master. Today, I'll tell you how to apply everything that you’ve learned so far and actually crack the interview. So, buckle up. Let’s go!

Read time: under 4 minutes

DON’T MISS THE INTERVIEW DAY

Here’s how the standard recruitment process looks like at most tech companies:

1. **Recruiter Screen:** After reviewing your resume, the recruiter will reach out to make sure that you’re the right candidate for the job.
2. **Online Assessment:** If the recruiter thinks you’re the right fit, you might get an online test which would have questions similar to Leetcode. Some companies don’t do online assessments and directly go to the phone screen.
3. **Phone Screen:** This round used to happen on phone but happens over Zoom these days. This is your standard coding interview with an engineer where they ask you Leetcode style question(s).
4. **Onsite Interviews:** This is typically a day with back to back interviews. It used to happen in the office but happens virtually these days. This typically contains 2-3 Coding Interviews, 1 Behavioral Interview and 1 System Design Interview. Sometimes there’s a Hiring Manager round as well. The mix might vary depending on the company, the team and the level.

Coding Interview

5 things to keep in mind in a Coding Interview:

1. Always ask **clarifying questions** before starting the solution to make sure you’re on the same page as your interviewer.

2. Walk through a **test case** before starting on the solution. This will make sure you are on the same page as your interviewer.
3. Don’t just dump all your **messy code** in one big function. Make sure your functions are modular whenever possible.
4. Don’t name your variables **x, y and z**. This is an easy giveaway of an inexperienced programmer. Give meaningful names to your variables to improve readability.
5. Think from **first principles**. The interview problem might be slightly different from the one you've solved.

Behavioral Round

Behavioral round is the interview where they ask you questions like “Tell me about a time when you failed at meeting a deadline” or “Tell me about a time when you had a conflict with a co-worker”. These rounds are designed to test your soft skills. Prepare your stories for common behavioral questions. Make sure your stories follow the STAR method.

System Design Interview

System Design Interviews are usually done for mid to senior level roles. But, it doesn’t hurt to be prepared. In System Design interviews, they ask you questions like “Design the architecture of a Hotel Reservation system” or “Design Whatsapp”. To answer these kinds of questions, you need a good understanding of Databases, Caching, Networking etc. Here are a couple of ways to prepare for these interviews:

1. Grokking the System Design Interview course
2. Designing Data Intensive Application book

HIDDEN TREASURE

Nothing can stop you. You keep finding the Hidden Treasure. If you want the solutions to the Top 10 most asked System Design problems, you can unlock the treasure here:

Refer to Unlock

📌 Or copy and paste your referral link to others: https://instabyte.io/subscribe?ref=A6lbQCYndC

Share your referral link on LinkedIn or with your entire class to unlock the treasures quicker.

WHAT’S NEXT

Most people think that the job search is over once you’ve got the offer. That could not be further from the truth. Winners always negotiate. On Day 5, I’ll share the salary negotiation techniques I used to increase my salary by $90,000.

If you’ve come this far, you must be enjoying the course. Follow me on Linkedin for more interview tips and tricks.

Can’t wait for tomorrow!

Cheers,
Sahil

228 Park Ave S, #29976, New York, New York 10003, United States

Gmail

Neeraj Dhakad <neerajbamboriya2@gmail.com>

Day 5: Never leave money on the table

1 message

InstaByte <newsletter@mail.interviewmaster.io> Sun, Feb 9, 2025 at 8:32 PM
Reply-To: InstaByte <newsletter@mail.interviewmaster.io>
To: "neerajbamboriya2@gmail.com" <neerajbamboriya2@gmail.com>

INTRODUCTION

You made it! Welcome to Day 5 of Interview Master.

There are 2 kinds of people in the world. The ones who don’t negotiate and the ones who get a better offer. Let’s make sure you’re the latter.

Read time: under 4 minutes

NEGOTIATION NUGGETS

Before we get into negotiation techniques, it’s important to know how tech compensation works. There are usually 4 components in a tech offer:

1. **Base salary:** What you get every month in your account.
2. **Sign-on bonus:** What the company pays you to leave your job and sign an offer with them.
3. **Annual bonus:** What the company pays you at the end of the year as bonus. It’s usually a percentage of your base salary.
4. **Stocks:** Companies can also give you their stocks at regular intervals after joining. This can be done using RSUs or Stock options.
5. **Relocation:** What company pays you to move for the job.

Your ability to negotiate is directly tied to your ability to say “no” to the offer. That’s why, you’ll have more leverage in negotiation if the other party thinks that you can say “no” to their offer. Here are a couple of things that can make them think that:

1. If you have more than one offer, they know that you can go elsewhere.
2. If you work at a better company than the one making you an offer, you might choose to stay back.

Getting multiple offers is by far the best way to have the upper hand in the negotiation. When I joined Google, I had offers from Meta and Uber as well. These offers were the main reason I could negotiate hard.

Having said that, I've helped people negotiate even when they didn't have another offer. My approach is very simple. The best way to do it is to show them that you're overqualified for the role. And that's why you deserve more. Go back to the job post and your notes from the interviews. Document everything that makes you look better than what they asked for. Prepare 3-4 points to make your case.

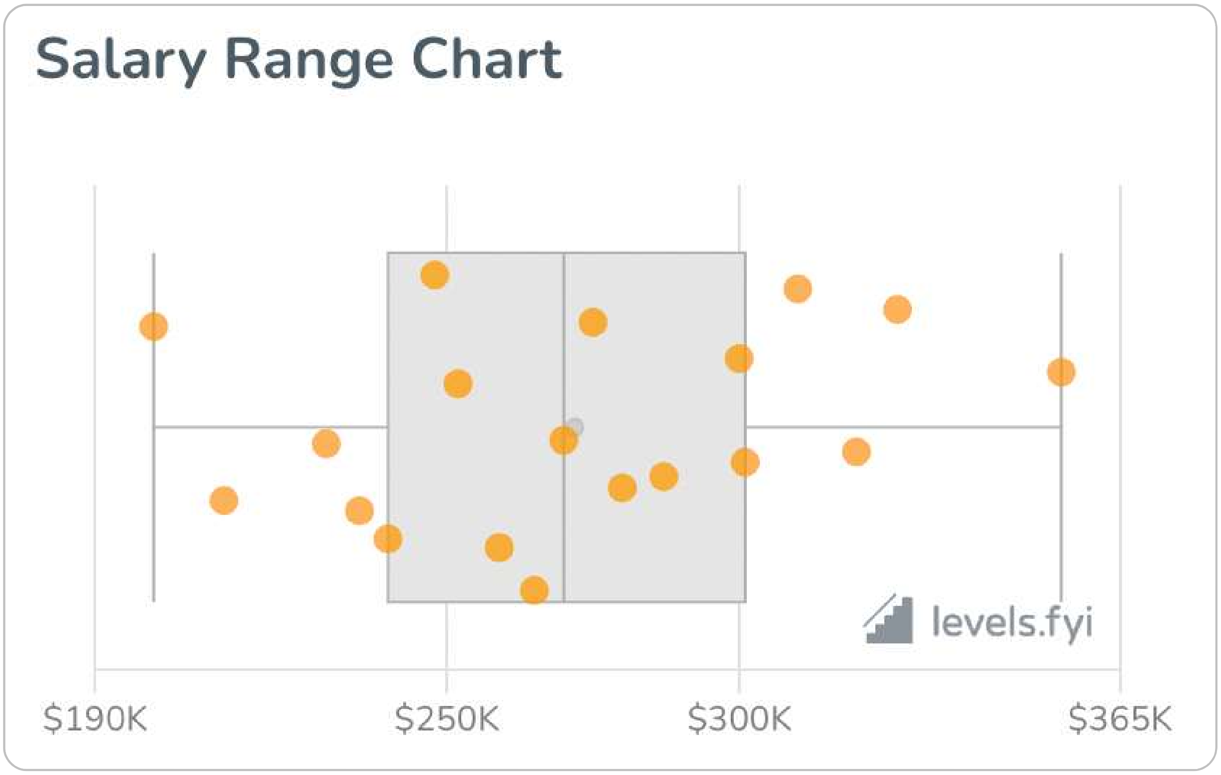
What to negotiate?

Here is the order of "ease of negotiation" for different salary components:
Stocks > Sign-on bonus > Base salary > Relocation > Annual bonus

It's very rare to negotiate relocation and annual bonuses. I always focus on the first three.

How much to negotiate?

Levels.fyi is the best resource to find out how much to ask for. They usually give a box plot for every level. I usually aim for the 90th percentile.



3 MISTAKES TO AVOID

1. **Not negotiate:** Even if the recruiter says the offer is non-negotiable, always negotiate.
2. **Give the number first:** Never give your expected salary or current salary. Listen to their number first.

3. **Worry:** It takes a lot of time for the company to find the right candidate. It's very rare for offers to be rescinded. Stay calm and negotiate.

WHAT'S NEXT

Congratulations! You've become the Interview Master. I can't wait to see what you achieve with everything you've learned in this course. If you found this course helpful, share it with your network on Linkedin and tag me.

I'll be back!

Cheers,
Sahil

228 Park Ave S, #29976, New York, New York 10003, United States