

Acing Tech Interviews: From Startups to Big Tech

Jason Bell & Christopher Mollard



Scan for Slido

Agenda

- Introductions
- Preparing for Behavioral & Competency Questions
- Preparing for Technical Exercises
- Congrats on Your Offer - Now What?
- Book & Podcast Recommendations
- Q&A

Slido: Scan & Add
Your Questions



Jason Bell

- 16+ years of experience across technology and consulting
- Tech commercial leadership roles at Amazon and three startups (one of which was acquired by Google)
- Qualified Amazon Hiring Bar Raiser, with a hiring veto, for 4+ years; led 500+ Amazon interviews
- Tech Careers Advisor for top UK & European Universities
- Current: Director of Know Your Pace, Personal & Professional Development



Christopher Mollard

- 12+ years of experience in UX as in house, consultant and contractor
- Lead UX and Scrum Master on numerous public sector projects
- Successfully delivered technology transformation on a small and large scale, working in and with development teams in Agile
- Early adopter and Thought Leader of UX in Agile
- Current: Director at J4G Design: UX, web and graphic design agency

Preparing for Behavioural & Competency



Behavioural & Competency

- Utilize STAR Framework
 - **S**ituation: provide necessary background
 - **T**ask: clarify your responsibility
 - **A**ction: succinctly describe what you did
 - **R**esult: explain the outcomes with a focus on impact you drove
- Common mistakes to avoid
 - Lengthy background
 - Forgetting **T**ask (your responsibility) and jumping to **A**ction after **S**ituation
 - Results which could have been achieved by a team without you



Behavioural and Competency Example

- People often say the simplest solution is the best. Tell me about a particular complex problem you solved with a simple solution.
 - This is testing how you innovate
 - **S:** When building a product that the client were still in the process of branding, we were continually asked to make content and brand updates
 - **T:** I was working on the front end, primarily with the content and styling
 - **A:** I implemented centralised components for wording and elements
 - **R:** This saved our delivery team hours a week, the equivalent to a day every fortnight which freed us up for more valuable and new work



Behavioural and Competency Example

- When have you taken on something significant outside your area of responsibility? Why was it important? What was the outcome?
 - This is testing your ownership
 - **S:** Working on a Scrum Agile delivery team, we would have daily stand ups to update the team on what we were working on
 - **T:** I had to discuss what I did yesterday; was doing on that day and whether there are any blockers. Another member of the team was blocked and couldn't continue with a front end component
 - **A:** Having heard this, I spoke up and said that I could take on the component that was blocking him
 - **R:** I resolved the issue quickly and my peer was able to pick up another piece of work. The team completed the Sprint on time and showed valuable progression for the client.



Preparing for Technical Exercises



Three Main Types of Technical Exercises



Coding:

- **Specific language proficiency:** Write code in a specific language like Python, Java, or C++ to solve a problem
- **Data structures and algorithms:** Implement a specific data structure like a binary search tree
- **Object-oriented programming:** Design classes and interfaces for a given scenario



Problem-solving:

- **Algorithmic puzzles:** Solve logical puzzles with code
- **Technical brainstorming:** Discuss potential approaches to a real engineering problem
- **Debugging:** Analyze and fix errors in existing code, demonstrating understanding of common pitfalls
- **Code review:** Review someone else's code and provide constructive feedback



Systems Design:

- **Scalable:** How to handle millions of users and requests
- **Real-time:** How to process data and respond in real-time, with low latency
- **Fault-tolerant:** How to tolerate failures and remain operational
- **Security-focused:** How to secure against common attacks



Hiring in Tech - Latest News

- The balance of power has shifted back to tech employers
- While a minority of experiences, watch out for software engineering interview processes that feel like a multiple-day work assignment
- For challenging technical interviews, prepare with:
 - [HackerRank](#)
 - [LeetCode](#)



Tech Job Interviews Are Out of Control

Tech companies are famous for coddling their workers, but after mass layoffs the industry's culture has shifted. Engineers say that getting hired can require days of work on unpaid assignments.



Preparing with HackerRank

The screenshot shows the HackerRank website interface. At the top, there is a navigation bar with links for Candidates, Tests, Interviews, Library, and Pricing. A search bar is located on the right side of the navigation bar. Below the navigation bar, the page title is 'Library > Library - HackerRank'. There are tabs for 'HackerRank', 'My Company', 'Archived', and 'Leaked'. A 'Create Questions' button is visible in the top right corner.

The main content area is divided into two columns. The left column contains a sidebar with a menu icon and a list of filters. The first filter is 'Status' with a sub-filter 'Not Leaked'. Below that is 'AI Solvable' with a sub-filter 'False'. The second filter is 'Binary Tree Search' with a difficulty level of 'EASY'. Below this is 'Coding' with a recommended time of '13 mins'. The third filter is 'Approximate Matching' with a difficulty level of 'MEDIUM'. Below this is 'Coding' with a recommended time of '33 mins'.

The right column displays the details for the 'Binary Tree Search' problem. The title is 'Binary Tree Search'. There are icons for '+', 'View', 'Try', and 'Insights'. The skills listed are 'Problem Solving (Intermediate)'. The recommended time is '13 mins', points are '50', and there are '14 test cases (5 samples)'. The coding difficulty is 'EASY'. There are tags for 'Interviewer Guidelines', 'Problem Solving', 'Algorithms', 'Binary Trees', and 'Data Structures'. The languages supported are 'C, C++, C++14, C#, Java 7+ 7 more'.

The description of the problem is as follows:

In a binary search tree, each node holds a *value* and a reference to as many as 2 *child* nodes, or *children*. The *root* node has no ancestors. The children are called *left* and *right*, and *subtrees* rooted at *left* and *right* are the left and right subtrees. If each node is considered the root of a subtree, each node value in its left subtree must be less than its own value. Likewise, each node in its right subtree must have a greater or equal value to the root. This allows for efficient searching.



Preparing with LeetCode

Explore Problems Contest Discuss Interview ▾ Store ▾

🔔 0 👤 Prem

◀ Back to Explore

☆ Favorite

Easy Collection

Top Interview Questions



Overview

This is LeetCode's official curated list of Top classic interview



Array

Array type of questions were asked in interviews frequen



Strings

String type of questions were asked in interviews frequen

Introduction



This is LeetCode's official curated list of Top classic interview questions to help you land your dream job. Our top interview questions are divided into the following series:

1. [Easy Collection](#)
2. [Medium Collection](#)
3. [Hard Collection](#)



Congrats on Your Offer - Now What?

- **Package Negotiation:**
 - a. Base salary
 - b. Bonuses
 - c. Shares vs Options: key to understand the vesting schedule and potential value
 - d. Pension
 - e. Other Financial Benefits like health insurance, number of holiday days
- **Non-Financial Elements:**
 - a. Career Development & Training
 - b. Work-Life Balance
- **Final Offer Tips:**
 - a. **Multiple Offer Leverage:** Aim to have interview processes with multiple employers end around same time
 - b. **Watch Out:** Be Careful to strike the right balance - while engineers are in high demand, if you are too demanding you may regret it (e.g. offer rescinded, ruined dynamic with future manager)



Book & Podcast Recommendations

Books

- **The Lean Startup** (Eric Ries)
- **Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days** (Jake Knapp)
- **The Inmates Are Running The Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity** (Alan Cooper)
- **Don't Make Me Think: A Common Sense Approach to Web Usability** (Steve King)
- **Hooked: How to Build Habit-Forming Products** (Nir Eyal)
- **How Google Works** (Eric Schmidt)

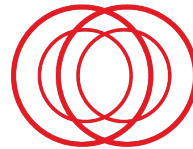
Podcasts

- **Catalyst Podcast Launch** (NTT Data)
- **Design Better: Intersection of Technology & Design** (Eli Woolery and Aaron Walter)
- **Web History** (Jay Hoffman & Jeremy Keith)
- **The Stack Overflow Podcast** (Eira May)
- **Hard Fork** (New York Times)
- **20VC** (The Twenty Minute VC)



Q&A

Scan for Slido



KNOW YOUR PACE
PERSONAL & PROFESSIONAL
DEVELOPMENT