



Interview Questions

w/ a focus on technical interview

GPI Extratation | 9/25

By: Keiffer Acoba, Amy Liu, and Jeremy Wang



A meme featuring Dwight Schrute from the TV show 'The Office'. He is shown from the chest up, wearing his signature light blue dress shirt and glasses. He has a serious, somewhat skeptical expression on his face. The background is a blurred office setting with another person in a light blue shirt visible in the background. The text is overlaid in large, white, bold, sans-serif font with a black outline.

**AM I READY FOR THIS
INTERVIEW?**

**FALSE. THE QUESTION IS, IS THIS INTERVIEW
READY FOR ME?**

Outline

- What is a technical interview?
- Different types: phone screens vs on-campus vs onsite vs coding challenge
- How to even get a technical interview?
- Techniques
- Examples
- Behavioral questions: how to prepare & examples
- Questions to ask the interviewer
- After the interview
- How to practice, followed by some Leetcode example

What is a technical interview?

- You will be expected to write some code, usually on a whiteboard or a notebook
- The problem will usually have several reasonable solutions of varying time and space complexity, and your goal is to eventually arrive at the optimal one
- They usually test purely algorithmic knowledge, but sometimes they include domain knowledge like details of C++ syntax (if you're applying to a C++ role)
- They will also sometimes require writing a lot of code very quickly (Citadel and Jane Street love to do this)

Sub-Types of Technical Interview

- **The algorithms interview (Most common):**
 - “Here is a problem that has many algorithms with different complexities. Find the best one and implement it”
- **The do-you-know-this-language interview:**
 - “Please explain what the Virtual keyword does in C++”
- **The write-a-lot-of-code-quickly interview:**
 - “Here is a (somewhat open-ended?) problem with low-algorithmic content, please write a reasonable solution and explain your design”
 - “Implement Tetris”
- **The domain-knowledge interview:**
 - “Here is a problem from our industry, explain how you would solve it under these constraints”
 - May or may not involve code

There's also ... (that you likely will not encounter until new grad so we won't cover today)

- System design questions
 - How would you design a color picker? (Barely any code, but talk through the UI, database, API, etc with the interviewer)
- Bug squashing
 - Given already implemented functions, find bugs and fix them
- Integration-style questions
 - Given an API, make calls to it to solve the problem
- & more

General Interview Process

1. Hackerrank / Online Coding Challenge
2. Technical (and Recruiter) phone screens OR on-campus
3. Onsite interview
 - “Onsite” but most are remote nowadays
 - 4-5 hours on zoom :) ← though internship interview processes are often shorter
4. Offer
5. Profit

How to even get a technical interview opportunity?

- For Hackerranks / coding challenges, some companies give them out almost automatically upon applying. If you want practice, APPLY APPLY APPLY!
- Apply EARLY!
- Go to career fairs / company events / stands / chats / etc and talk to recruiters or engineers.
 - “Apply online” but you could get lucky
- Get referrals from your friends
 - Google, Apple, Facebook, Microsoft, Amazon
 - Upperclassmen are your friends here

Techniques

- Get a good night of sleep & eat properly

Sleeps for one hour

Fails interview



Techniques

- The flow
 - Clarifying questions
 - Brainstorm data structures before coding
 - Think of the brute force solution first
 - Tell the interviewer a high-level overview of your solution before coding for real
 - State that you know this is too long / inefficient / blah, state why
 - Now try to improve upon it
 - You can ask for hints if you're absolutely stuck!

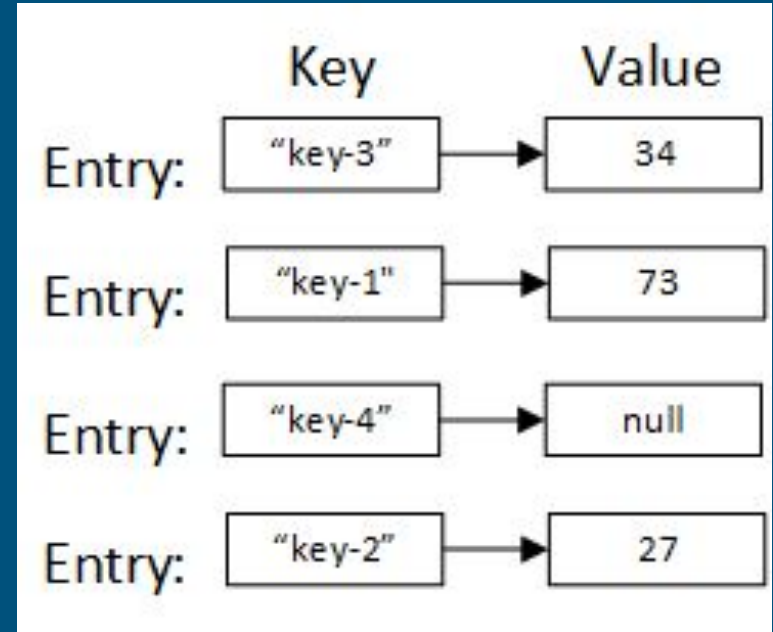
Techniques

- Think out loud
 - Interviewer is more concerned with your thought process in achieving the answer than correctness of the code
 - More than 10 seconds of silence **without progress** is bad
 - Even if you think you'll sound dumb, talk out loud anyways! You might find yourself catching mistakes and fixing them as you talk (woo rubber duck method)
- Incremental difficulty
- Utilize constraints
 - Is the input already sorted? (If not, does sorting it help?)
 - Are input integers only positive?

Map - A magical data structure



- hash map, hash table, dictionary, etc.
- Lets you store (key, value) pairs and read them in (usually) $O(1)$ time!
- Even if you haven't learned data structures yet, try to learn how to use this one (don't need to know how exactly it's implemented yet)
- set: map with only keys, useful on duplication-related questions
- **does not preserve ordering**



Example: Intersection of 2 arrays

- You have two sets, represented as arrays with no duplicates. The arrays are small enough to fit in memory. Find the set-intersection of the two arrays (the elements present in both).
 - Store the elements of one array in a hash table.
 - Loop over the elements of the second array, outputting anything that was already in the hash table.
- How can we use less memory when the arrays are sorted?
 - Keep an index into both arrays. While neither index is at the end of the array, compare the elements at those indices. If they are the same, output that value and increment both indices. Otherwise, increment the index whose array element was smaller.

Example: The Virtual Keyword in C++

- In my experience this is the most asked interview question for C++ roles
 - I have answered it at least three times for three different companies
 - People ask it because it is a good separator between people who actually know C++ and people who only know C.
- You have two classes, A and B, where A inherits from B, and you construct an object obj of type A but cast it to type B. Then you call method obj.foo().
 - If foo() was virtual, then A::foo() would be called.
 - If foo() was non-virtual, then B::foo() would be called.
- For non-virtual functions, C++ bakes in the implementation from the “static type” of the object, whereas virtual functions call into a lookup table that retrieves the implementation for the object’s “dynamic type”.
 - This behavior is nicer, but has a runtime cost and so should be avoided for performant code.



Behavioral questions

“So, tell me about a time where you had to resolve a conflict within a team.”



Behavioral Questions

- While there are dedicated “behavioral” rounds where you usually talk to a manager about these things, sometimes during technical interviews, they slip it in
- Meant to assess your work style, collaboration and communication skills, leadership potential, how you handle challenges, see if you ~vibe~ with company values, etc
- Bring depth to your resume and character
- Answer with stories of past action, not “I would have approached it...”

Example questions (from [indeed.com](https://www.indeed.com), this nice behavioral interview [presentation](#), and past experiences)

- Tell me about a time you made a commitment and did not deliver it.
- Tell me about a time you made a mistake and how did you recover from it.
- How do you respond when you disagree with a coworker?
- Tell me about a time you had multiple responsibilities to manage. How did you respond to this situation?
- Can you give me an example of how you set goals for yourself?
- Tell me about a time you were new to a situation or environment. How did you adapt?
- Tell me about a time you needed information from someone who wasn't responsive. How did you handle it?

Prepare the story... (1/2)

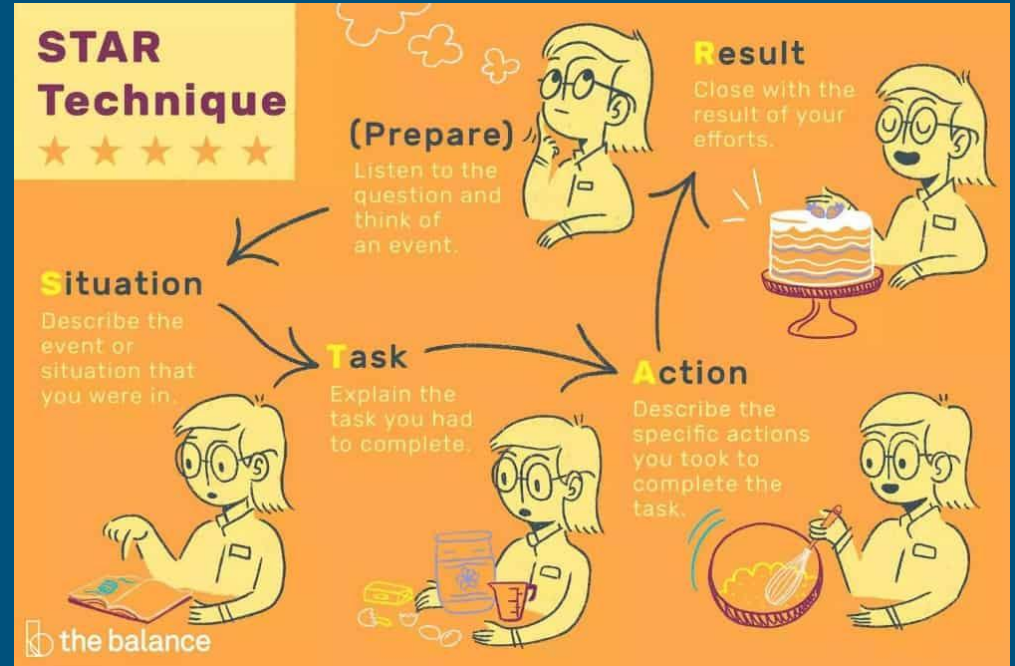
- For every experience on your resume (+ personal projects, course projects, extracurricular activities) come up with *something specific* to talk about.
 - Challenge you ran into (how did you overcome it?)
 - team collaboration issue (how did you resolve it?)
 - didn't meet the deadline (and how did you recover or revise the timeline?)
 - decisions you made (why? How? Did you receive opinion from others?)
 - Persuading others (why did they need persuading? How did you do it?)
- Perseverance & Learning from the experience

Prepare the story... (2/2)

- RESEARCH THE COMPANY!!
- Prepare stories that align with *each* of the company values
 - Customer Obsession
 - Ownership
 - Invent and Simplify
 - Etc ...

Tell the story...

- Make sure your story answers the question
- Use “STAR”
- Focus on “what I did” instead of “what we did”
- Don’t ramble -- 2 mins max



Example: How do you respond when you disagree with a coworker?

- [Situation] I worked with one other partner in a 36-hour hackathon project. I worked on the backend while my partner worked on the front end.
- [Task/Conflict] My partner was spending too much time trying to change the color of a link when the user hovered over it. However, I thought that as long as the link works, we should move on and aim for something working and presentable.
- [Action] I asked for his reasoning, which was wanting to be user-friendly and give an indication when hovering. And I told him my thoughts as well. I asked my partner to step through some code and elaborate on why it was hard to change the color. I tested out some code myself and found that underlining the link was very easy to do. So, I proposed a compromise to implement an underline when the user is hovering the link, so that while it's not as aesthetic as a color change, we will still give the user an indication, but also leave enough time for the rest of the project.
- [Result] My partner agreed to the compromise, and we were able to create a working prototype to the end to show the judges.

Quick slide about recruiter phone interviews

- 15-30 mins. Super chill
- Present yourself as a nice person that's interested in the company
- Recruiter will likely ask you
 - Logistical questions (when can you work, are you authorized, etc)
 - Why are you interested in the company
 - What do you look for in an internship / what you want to get out of an internship (mentorship, new skills, put school knowledge to work, etc)
- Will tell you more about rest of the process
- Great time to ask questions! (and to also show you are interested)

Questions to ask the interviewer

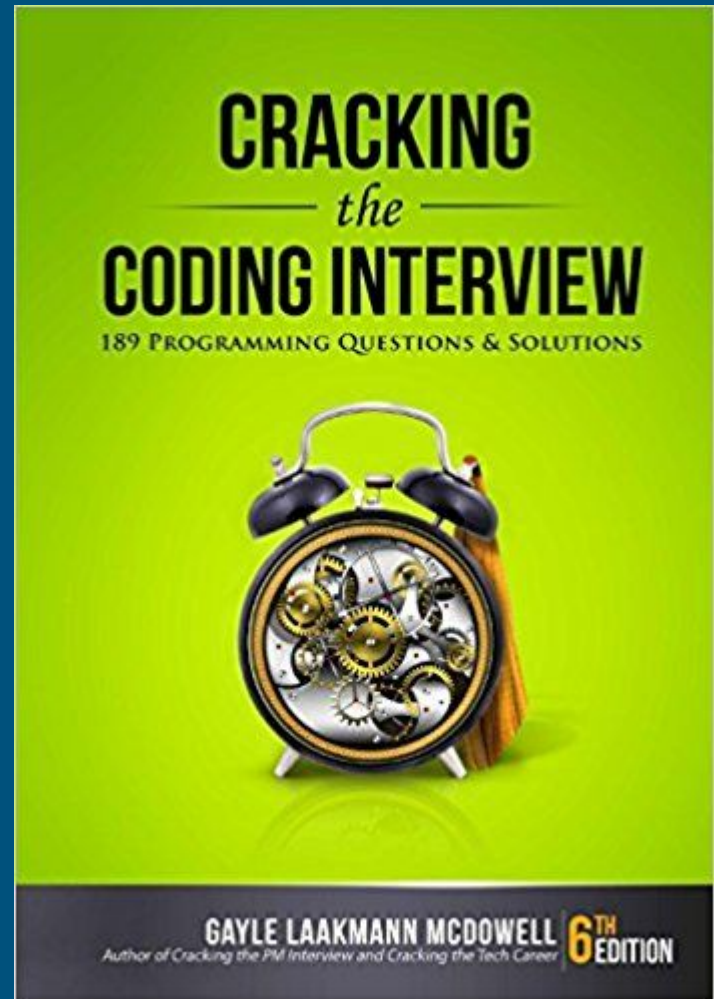
- Name, job title, maybe email (for reference later on)
- What made you choose company X?
- What's the most satisfying project you've worked on?
- What's a typical day for an intern like?
- Any example projects interns have worked on?
- What's your favorite thing about working for your company?
- How does this company compare to other places you've worked before?

After the interview

- Thank them for their time!!
- (If you didn't bomb it) and didn't hear back in 1 weeks, consider following up with the recruiter / interviewer!

How to practice

- Cracking the Coding Interview
 - This book is amazing, seriously
 - Start here if leetcode is too hard right now
- [Leetcode](#)
- Rubber ducky method
 - Vocalize, vocalize, vocalize!
- Visit Kevin Collins or Career Center for tips or mock interviews!
 - [CPDC website's resources](#) (scroll down and click on SCS - undergrad)



Leetcode time!

- A. [two-sum](#)
- B. [valid parentheses](#)
- C. [palindrome](#)
- D. remove duplicates from an array
- E. [climbing stairs](#)
- F. [counting bits](#)
- G. [product of array except itself \[MEDIUM\]](#)