How to Prepare Your Resume (and Yourself!) for Technical Interviews

Today

- Choosing Companies
- Resumes
- Before the Interview
- During the Interview
- After the Interview

Choosing Companies

Questions to Ask Yourself

- What are you looking for out of an internship / full-time role?
- Would you be excited to get up every day and work on this project?
- Would you be excited to get up every day and work with this team?
- Do your values align with the company’s values?
- Does the company size fit with what you’re looking for?

Questions to Ask Yourself

- **What are you looking for out of an internship / full-time role?**
- Would you be excited to get up every day and work on this project?
- Would you be excited to get up every day and work with this team?
- Do your values align with the company’s values?
- Does the company size fit with what you’re looking for?

Questions to Ask Yourself

- Everyone is different
  - Impact?
  - Learning?
  - Work-life balance?
  - Location?
  - Money?

Questions to Ask Yourself

- What are you looking for out of an internship / full-time role?
- **Would you be excited to get up every day and work on this project?**
- Would you be excited to get up every day and work with this team?
- Do your values align with the company’s values?
- Does the company size fit with what you’re looking for?

Questions to Ask Yourself

- Look at where the company & product is today
- Learn about where the company & product is going
  - This can be even more important

Questions to Ask Yourself

- What are you looking for out of an internship / full-time role?
- Would you be excited to get up every day and work on this project?
- **Would you be excited to get up every day and work with this team?**
- Do your values align with the company’s values?
- Does the company size fit with what you’re looking for?

Questions to Ask Yourself

- Ask questions to your interviewers!
  - This is useful for you, not just to look good
- Once you have an offer, ask to talk to prospective team members
  - Chances are, your recruiter will offer!

Questions to Ask Yourself

- What are you looking for out of an internship / full-time role?
- Would you be excited to get up every day and work on this project?
- Would you be excited to get up every day and work with this team?
- **Do your values align with the company’s values?**
- Does the company size fit with what you’re looking for?

Questions to Ask Yourself

- Every company has a different set of values
  - Defined as “what this company values more than other companies”
- What do you value?
  - Collaboration?
  - Growth?
  - Scrappiness?
  - Stability?

Questions to Ask Yourself

- What are you looking for out of an internship / full-time role?
- Would you be excited to get up every day and work on this project?
- Would you be excited to get up every day and work with this team?
- Do your values align with the company’s values?
- **Does the company size fit with what you’re looking for?**

	Small (0-50)	Medium (50-500)	Large (500+)
Your role	Many roles	Generalist	Specialist
Autonomy	Do what needs to be done	High level mandate	Clear direction
Product influence	Eng/pm hybrids	Involved in planning	Receive plans
Agency	Choose (or start) your team	Choose among some teams	Assigned a team
Access to information	Everything is discussed openly	Lots of information available	Information carefully controlled
Mentorship	Ad-hoc	Mentors, buddies, bootcamps	“XYZ University”
Financial return	Low salary + very high potential upside	Reasonable salary + high potential upside	Very high salary + low potential upside

Resumes

Not that important

30 seconds

10 Resume Rules

1. One page, no exceptions
2. Make it easy to skim
3. Make contact info obvious
4. Highlight specific accomplishments
5. Include interesting personal projects

10 Resume Rules

6. No charts or ratings
7. No objective
8. Use a professional email address
9. Include relevant links: LinkedIn, GitHub, portfolio
10. Don’t sweat aesthetics

Emily Saavedra

88 Hendford Hill, London B22 0GX, United Kingdom | 078 3515 0056 | emilysaavedra@gmail.com

Profile

Highly accurate and experienced Data Scientist adept at collecting, analyzing, and interpreting large datasets, developing new forecasting models, and performing data management tasks.
Possessing an extensive analytical skills, strong attention to detail, and a significant ability to work in team environments, Emily is presently looking for a Data Scientist position with a forward-moving company.

Work experience

09/2017 – 02/2019
LONDON, UNITED KINGDOM
SpyBiotech, Inc.
Data Scientist

- Assisted in scientific research on DNA cloning and analyzed the results.
- Collected, studied, and interpreted large datasets; conducted reports; performed accurate, successful data management.
- Developed and implemented new forecasting models which increased company productivity and efficiency.
- Participated in monthly meetings with executives, provided information on the progress.

07/2016 – 09/2017
LONDON, UNITED KINGDOM
Data Scientist
CGL, Inc.

- Collected, analyzed, and interpreted raw data from various websites.
- Collaborated with the Operations and Technology Department on the development of new automated data management/analysis software which increased the overall productivity and cut unnecessary costs.
- Maintained and managed company's MS SQL server.
- Increased the accuracy of forecasting software from 80% to 95%.

Education

09/2012 – 05/2016
CHICHESTER, UNITED KINGDOM
Mathematics and Statistics
University of Chichester
First Class Honours
Clubs and Societies: Business Club, Golf Club, Riding Club

09/2010 – 05/2012 EL PARAISO, SPAIN
IB Diploma Programme
The International School Estepona
Graduated with Distinction (Grade 1 - A/excellent equivalent in all 6 subjects)

Skills

LANGUAGES

Spanish	**Native**
English	**Full**
French	**Limited**

COMPUTER/DATA ANALYTICS SKILLS

Microsoft Office	
MS SQL Server	
Sisense, Zoho Analytics	
GoodData, Qlik Sense	

INTERPERSONAL SKILLS

Accuracy	
Analytical Skills	
Detail Orientation	
Good Team Player	
Multitasking	

Volunteering

06/2014 – 08/2014 SUVA, FIJI
English Tutor
Go Overseas

Certificates

10/2016
Professional Certificate in Data Analysis
The Chartered Institute for IT

MARISSA MAYER

Business Woman & Proud Geek

mmayer@yahoo-inc.com
https://marissamayr.tumblr.com
Sunnyvale, CA

EXPERIENCE

President & CEO

Yahoo!

07/2012 - Ongoing Sunnyvale, CA

- Led the $4 billion acquisition of the company by Verizon
- Acquired Tumblr for $1.1 billion and moved the company's blog there
- Got to $1.6 billion in GAAP revenue in mobile, video, and social
- Tripled the company's mobile base to over 600 million active users

Vice President of Location & Local Services

Google

10/2010 - 07/2012 Palo Alto, CA

- Positioned Google Maps as the world leader in maps and navigation

Vice President of Search Products & UX

Google

2005 - 2010 Palo Alto, CA

Product Manager & Technical UI Lead

Google

2001 - 2005 Palo Alto, CA

- Optimized usability on Google's homepage to the smallest detail

Product Engineer

Google

1999 - 2001 Palo Alto, CA

- Joined the company as employee #20 and female employee #1

EDUCATION

M.S. in Computer Science

Stanford University

1997 - 1999

B.S. in Symbolic Systems

Stanford University

1993 - 1997

MOST PROUD OF

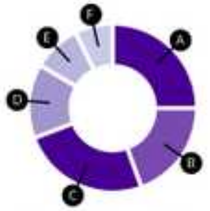
Courage I had
to take a sinking ship and make it float

Persistance & Loyalty
I showed through hard times at Yahoo following its acquisition

Google's growth
from 100k daily searches to 1 billion+

Inspiring women in tech
by being the youngest CEO on Fortune's list of the 50 most powerful women

MY TIME



- A Spending time with my children
- B Publicly resolving Yahoo! investor issues
- C Showing Yahoo! employees their work has meaning
- D Building a biz-dev strategy for Yahoo's future after the Verizon acquisition
- E Serving on the boards of New York and San Francisco ballet companies
- F Creating spreadsheets for my amazing cupcake recipes

Before the Interview

What to Expect

Timing

- Start interviewing as early as you can
  - Common to interview in late summer / early fall
- Plan interviewing into your schedule
  - It takes a lot of time
- Look for opportunities geared at your experience
  - Larger companies have apprentice, first-year, etc. programs

Process

- Maybe: Online coding challenge (~30 minutes)
- Maybe: Take home (hours-days)
- Probably: Technical phone screen (45-60 minutes)
- Almost definitely: Onsite (3-5 interviews, 45-60 minutes each)

Types of Interviews

- Algorithms
- Coding
- Practical
- Systems Design
- Culture

Algorithm Interviews

- More about getting the algorithm than coding it up
- Problem is hard, but coding the solution is easy

Coding Interviews

- Problem seems simple, but implementation is complex
- Focus on getting the details right, bugs, corner cases

Practical Interviews

- Given an existing codebase, make changes to it
- Build something from scratch, using the language of your choice
- Much heavier emphasis on tools, debugging, workflow

Systems Design Interviews

- Technical design
  - “How would you architect the Gmail app?”
- Product design
  - “What’s the biggest feature missing from Gmail?”

Culture Interviews

- Looking back
  - What were some of the hardest problems you solved?
  - What would you do differently?
- Looking forward
  - Why this company?
  - What’s important to you?

Preparing for Interviews

Start with the Basics

- Practice decreases stress
- Choose a language, and stick with it as much as possible
- Learn syntax, builtins, and error messages

Start with the Basics

- Know how to:
  - Create a function
  - Define a class
  - Work with strings
  - Work with lists
  - Work with trees

Start with the Basics

- Interview problems rarely have complex complexity
  - Constant
  - Logarithmic
  - Linear
  - Polynomial
  - Exponential

Build your Toolbox

- Many problems reduce to a few core concepts:
  - Recursion, Divide-and-Conquer
  - Graph Searches
  - Greedy Algorithms
  - Strings
  - Searching & Sorting
  - Dynamic Programming

Build your Toolbox

- Tools you have at your disposal:
  - Arrays, linked lists
  - Hash tables
  - Binary search
  - Shortest-path algorithms
  - Memoization

Simulate the Environment

- Do lots of practice problems, talking out loud without Google
  - Time yourself, no cheating
  - Look for patterns: bad at recursion? Then do more recursion
  - Do practice interviews with friends
- Practice a variety of problems
- Then do more practice problems

During the Interview

Process, not output

The Icebreaker

- Interviews don’t just start with a coding problem
- Have a 1-3 minute introduction ready to go
  - Practice decreases stress
- Don’t ramble

Approaching Technical Problems

- Do not panic
- Always tell your interviewer if you've seen the problem before
- Ask clarifying questions
  - Do not make any unstated assumptions
  - Verify you agree on the output for an example input

Approaching Technical Problems

- Continually think out loud, even if it feels weird
  - Your interviewer cannot help you if you are silent
- Talk through multiple possible approaches
- If helpful, draw a diagram

Approaching Technical Problems

- Try to pattern-match against what you know
  - Can we formulate this as a graph?
  - Can we formulate this recursively?
  - Can we use a binary search?
  - Can we decompose and memoize sub-problems?

Approaching Technical Problems

- Think general before specific
  - Don’t write out a list of corner cases first
- If your solution seems too complicated, it probably is
  - Remember, these are designed to be solved in 30-45 minutes

Process, not output

Writing Code

- Constantly communicate your thought process
- Get something working first
  - It is always better to leave with something rather than nothing
- Then, simplify and optimize
- Don’t rush, but don’t assume the initial problem is the entire interview
- Listen to your interviewer—they want to help!

Writing Code

- Code quality matters!
  - Decompose into functions as needed
  - Use readable variable names
  - Factor out common logic

Process, not output

Testing Code

- Be the computer: run through the function verbally for some input
  - Show that you can reason about your code without running it
  - Proactively look for and fix bugs

Really Testing Code

- If you’re allowed to run your code on a computer:
  - Write test cases
  - Don’t just guess-and-check
  - Print/debug relevant state to fix issues

Process, not output

Example Problem

Prompt

Let’s define a rotated array as a sorted array where the numbers have all been rotated to the right some number of places, with numbers wrapping around when they reach the end of the list.

[1, 2, 3, 4, 5] rotated 3 times is [3, 4, 5, 1, 2]

Given a rotated array, find the number of times it was rotated.

Approaching the Problem

- We are not panicking
- First, let’s confirm we understand the problem by giving an example
  - So [2, 4, 6] rotated 2 times would be [4, 6, 2]? Okay great
- Next, ask clarifying questions
  - This list will only have sorted integers? Okay great

Approaching the Problem

- Let’s look at the example again
  - [1, 2, 3, 4, 5] rotated 3 times is [3, 4, 5, 1, 2]
- We want to find the number where both the left and right are larger
  - This is just the minimum of the array! (Is it? Try another example)
- We could use linear search
  - Sure, but can you do better?

Approaching the Problem

- We are still not panicking
- Toolbox time:
  - Recursion, Divide-and-Conquer
  - Graph Searches
  - Greedy Algorithms
  - Strings
  - Binary Search

Approaching the Problem

- We are still not panicking
- Toolbox time:
  - Recursion, Divide-and-Conquer
  - Graph Searches
  - Greedy Algorithms
  - Strings
  - **Binary Search**

Approaching the Problem

- Binary search works on a sorted list, and this list is kinda sorted
- We can just use binary search with a different search condition!

```
def count_rotations(rotated):
  low = 0
  high = len(rotated) - 1
  while low <= high:
   mid = (low + high) // 2
   if rotated[mid] <= rotated[high]:
      high = mid - 1
   elif rotated[mid] >= rotated[low]:
      low = mid + 1
```

```
def count_rotations(rotated):
  low = 0
  high = len(rotated) - 1
  while low <= high:
   mid = (low + high) // 2
   if rotated[mid] <= rotated[mid + 1] and
      rotated[mid] <= rotated[mid - 1]:
     return mid
   if rotated[mid] <= rotated[high]:
     high = mid - 1
   elif rotated[mid] >= rotated[low]:
     low = mid + 1
```

```
def count_rotations(rotated):
  low = 0
  high = len(rotated) - 1
  while low <= high:
   mid = (low + high) // 2
   if rotated[mid] <= rotated[(mid + 1) % len(rotated)] and
      rotated[mid] <= rotated[(mid - 1 + len(rotated)) % len(rotated)]:
     return mid
   if rotated[mid] <= rotated[high]:
     high = mid - 1
   elif rotated[mid] >= rotated[low]:
     low = mid + 1
```

```
def count_rotations(rotated):
  low = 0
  high = len(rotated) - 1
  while low <= high:
    if rotated[low] <= rotated[high]:
      return low
    mid = (low + high) // 2
    if rotated[mid] <= rotated[(mid + 1) % len(rotated)] and
       rotated[mid] <= rotated[(mid - 1 + len(rotated)) % len(rotated)]:
      return mid
    if rotated[mid] <= rotated[high]:
      high = mid - 1
    elif rotated[mid] >= rotated[low]:
      low = mid + 1
```

Writing Code

- Now, let’s run our code on the example input: [3, 4, 5, 1, 2]
- First we land on 5, doesn’t work
- 3 < 5, but 5 > 2, so we discard the left half
- Now we land on 1, which meets the condition!
- Answer is the index of 1, which is 3

Runtime

- Binary search is O(log n), so this must be too
  - Just have this down cold!
- Do one last check for bugs before we say we’re done

After the Interview

The Reverse Interview

- This is very important!
- Have questions ready in advance that are not Google-able
  - What does your typical day-to-day look like?
  - Tell me about your team culture
  - What would you change about your team?
  - What’s something you were proud of shipping recently?
  - What has your path at this company looked like?

Don’t Worry

- Interview performance is high-variance
- Everyone bombs interviews, so don’t get discouraged
- Each interview is an opportunity to practice and get better
  - Do a self-retrospective after each one

Takeaways

Takeaways

- Have the basics down
- Build your toolbox
- Do lots of realistic practice problems
- Communicate a solution before writing code
- Verbally run your code and look for issues

Process, not output

How to Prepare Your Resume (and Yourself!) for Technical Interviews