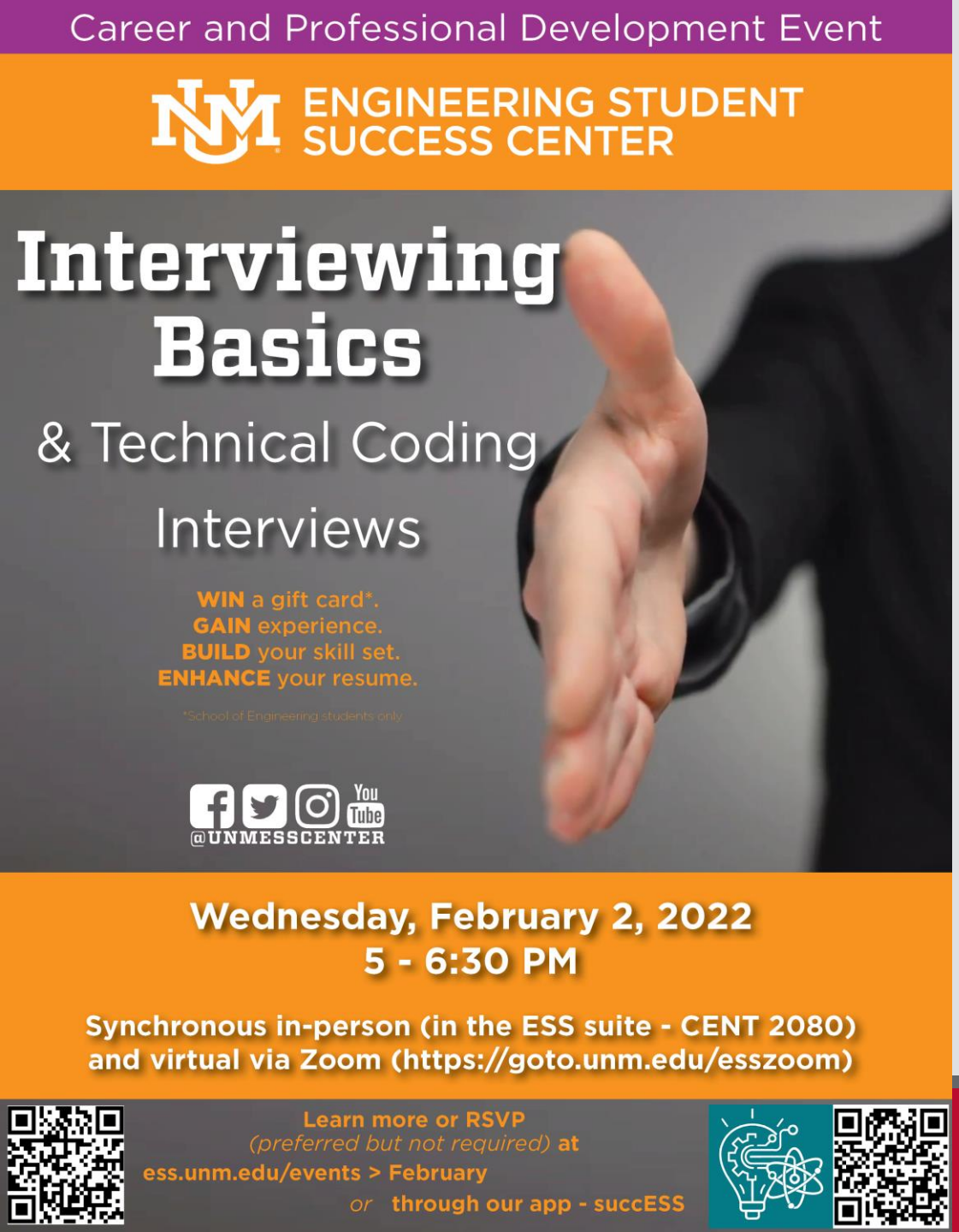


Interviewing Basics with Nada Abdelhack, GCDF

Technical Coding Interviews with Sahba Tashakkuri

ENGINEERING STUDENT SUCCESS CENTER

Effective Interviewing 101

PRESENTED BY:

NADA ABDELHACK, MA, GCDF

Who Am I?

- Internship & Job Placement Coordinator
- Resumes and cover letters
- Career Advisor = Tips for Career Success
- Who we are at ESS

ENGINEERING STUDENT SUCCESS CENTER

Prepping for an Interview

- Outfits
- About Us
- Review
- Elevator Pitch
- Be On Time

ENGINEERING STUDENT SUCCESS CENTER

Interviewing Anxiety

Prepare Properly

To-Do List

Use Visualization

ENGINEERING STUDENT SUCCESS CENTER

Relax

Physically Relax

Smile

Posture/ sit up straight

Pause

Take time to think

Speak slowly and calmly

Effective Interviewing

Eye contact

Good Posture

Firm Handshake

Breathing Techniques

No Fidgeting

Rehearse your Answers

Talk Slowly

Professional Dress

ENGINEERING STUDENT SUCCESS CENTER

One-on-One Interviews

Develop Rapport

Open-Ended Questions

80/20 Rule

ENGINEERING STUDENT SUCCESS CENTER

Panel Interviews

Introduce Yourself

Take Notes

Answer questions

Verbal/Non-Verbal Communication

Ask Meaningful Questions

Close Successfully

Send Thank you Letters

When Answering Questions

Interviewer’s perspective

Respond effectively and Completely

Take Notes

Collect your thoughts

Ask for clarification

ENGINEERING STUDENT SUCCESS CENTER

Virtual Interviews

No Distraction

Clutter free

Proper Lighting

Your Screen Name

Test Your Software

Dress Properly

Body Language

Your Documents

ENGINEERING STUDENT SUCCESS CENTER

Phone Interviews

Plan ahead

Speak clearly

Close the Call

Phone Interview Tips

- Posture
- Stand Up
- Speak Directly into the Phone
- Voice and Tone
- No Smoking, Chewing Gum, Eating, etc.
- Smile
- Energetic and Interested

ENGINEERING STUDENT SUCCESS CENTER

Your Accomplishments

Impact

Quantifiable

Goals Reached

Common Interview Questions

Tell me about your self.

Why do you want to work at this company?

Why do you want this job?

Why should we hire you?

What can you bring to this company?

What are your greatest strengths?

Why are you leaving your current job?

What are your Weaknesses?

Don’t mention weaknesses that are main requirements for the role

Show an employer what you’ve done to tackle your challenges

Use the STAR Method to tell your story

Avoid Cliches

STAR Method

Situation

Task

Action

Result

***R**elate the story back to the position.

Questions to Ask your Interviewer

Have I Answered all Your Questions?

What Will Be the Biggest Challenge for the Person Filling This Position?

Why is this Position Available?

What do the career paths of those who have held this position look like?

ENGINEERING STUDENT SUCCESS CENTER

Salary

When to Negotiate

Be Confident

Determine Your Worth

Multiple Level Interview

Non-Negotiable Positions

Things to Avoid During an Interview

Clueless About the Company

Talk too Soon About Money

Arriving Late

Forget Copies of Your Resume

Trash a Previous Employer

Lack Enthusiasm

Forget to Ask Questions

Talk Too Much

Leave Your Cell Phone On

What NEVER to Say at an Interview

"That's a great question!"	"What is the title of the role, again?"	"I've actually never done this type of job before, but…"	"I really can't imagine anyone more qualified than me."
"My last boss was *terrible.*"	"This will be a great stepping-stone to my next career move."	"I don't know."	"I don't have any questions for you."

In Summary


Ability
Acceptance
Work Ethics
Genuineness
Loyalty
Recommendations
Honesty
Communication Skills
Personality
Initiative
Sincerity

ENGINEERING STUDENT SUCCESS CENTER

Conclusion

Help with:

- resumes
- cover letters
- career advisement

Please reach out to me,

Nada Abdelhack at
esscareers@unm.edu

Questions?

Give feedback.

Win a gift certificate!

goto.unm.edu/ess-feedback

@UNMESSCENTER

Don't forget to follow up on social media.

Technical Coding Interviews
Why & How

Sahba Tashakori
Feb 2, 2022

My Background

- CS BS, MS (Dec 2020) at UNM
- Currently ML Engineer at Dell; will join Google in April
- Companies I successfully interviewed with:

Outline

Graphic from Financial Times

Part 1

- Intro
- Working at the big tech companies, pros & cons
- Overview of the coding interviews
- The philosophy behind coding interviews

Part 2

- How to prepare
- At the coding interview
- Don't we need to land an interview first!?
- Accepting/rejecting offers
- Once you start

Advantages of working at these companies

- Benefits
- Learning & Support & Growth
- Scale and `*impact*`
- Many cool projects to work on
- Future opportunities
- Job security
- Interview confidence and preparedness

Advantage: Working with Student Visas

- **The Response I got from Dell HR**
  - All international students are encouraged to apply
  - They work with OPT, CPT
  - They sponsor for H1B
- Similar for all other big tech companies
  - In particular Amazon, Microsoft, Apple, Google, Facebook (Meta)
- These companies have dedicated immigration teams

Why/Why not Final Thoughts

- Good option to have
- Try to learn what your preference is
- Aim for an internship to become familiar with the company
- Not particularly stressful

Outline

Part 1

- ✓ Intro
- ✓ Working at the big tech companies, pros & cons?
- ➢ Overview of the technical interview process
- ■ The philosophy of coding interviews

Part 2

- How to prepare
- At the coding interview
- Don't we need to land an interview first!?
- Accepting/rejecting offers
- Once you start

General Format (My Recent Experience)

My Recent (Dec & Nov 2021) experience (for full-time)

- Google
  - 1 online assessment (2 questions)
  - 1 phone interview (screening)
  - Final round: 5 interviews in 1 day

- Amazon
  - Online assessment (2 questions)
  - Work simulation and personality survey
  - Final round: 4 interviews
- Very similar for other companies
  - Also possible: system design, take home projects, IQ games …

The Questions

- Practical problems; you need to apply your CS knowledge to new situations and write the code to implement your ideas
- Good questions usually have multiple hurdles
- They are designed to make you get stuck
- Almost always there are multiple solutions to the problems
  - Which solution is better in what scenario?

The Core Idea

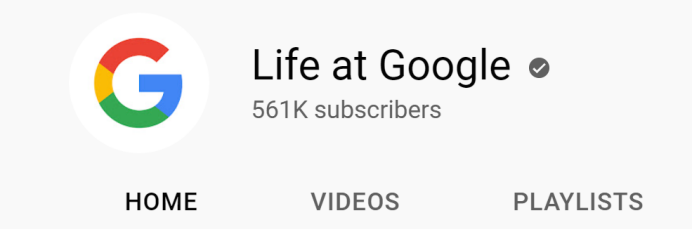
They want to hire you if you

- Have passion
- Took the time to learn the basics and practice
- Can think critically
- Can communicate
- Can be trained

Coding Interviews - A Quick Overview

In each coding interview session you will (in ~30-40 minutes):

- Discuss the problem and solutions
- Evaluate tradeoffs, time/space complexity of different solutions
- Code!
- Test your code!
- Follow ups (my favorite)



https://www.youtube.com/c/LifeatGoogle

What Do Coding Questions Look Like?

- Given a BST, find all of the arrays that would result in the same tree, if the array elements were to be inserted into the BST one by one (CCI)

- Given a long string and a shorter string, find all of the permutations of the shorter string in the longer one. (very good example for progression from inefficient solutions to the optimal solution: O(S!*L), O(S^2*L), O(S*L), O(S))

  Example: abdcabefbcadcbfgabcd

  target=abcd

- Given a robot in a grid that can turn CW and CCW and move one cell forward, write a program that guides the robot to clean the whole grid. There are blocked cells on the grid that you can only detect if you attempt to move into them. Return False if it is impossible to clear the grid without getting stuck. (previous Google question)

Outline

Part 1

- ✓ Intro
- ✓ Working at the big tech companies, is it worth it!?
- ✓ Overview of the coding interviews
- ➢ The philosophy of coding interviews

Part 2

- How to prepare
- At the coding interview
- Don't we need to land an interview first!?
- Accepting/rejecting offers
- Once you start

Technical Interview Debate

- Coding part is here to stay for now
- Basic knowledge, critical thinking, software dev skills, and communication
- Balances the field in favor of recent-grads & junior devs
  - Alternative: Have 'X' years of experience with technology 'Y', etc
- Good candidates can easily get eliminated (false negative)
- Sometimes not done correctly especially by smaller companies

Questions about the first part?

- ✓ My background
- ✓ Working at the big tech companies, is it worth it!?
- ✓ Overview of coding interview format
- ✓ The philosophy of coding interviews
- ➢ How to prepare
- ■ At the interview
- ■ Don't we need to land an interview first!?
- ■ Accepting/rejecting offers
- ■ Once you start

Coding Interview Format Overview Revisited!

You will:

- Panic! Get stuck!
- Discuss the problem and solutions (talk as you think) (if with a real person)
- Evaluate tradeoffs, time/space complexity of different solutions
- Get stuck!
- Code!
- Get stuck!
- Test your code! (conceptual, small test case)
- Find errors! Panic! Fix errors
- Follow ups (my favorite)
- Get stuck some more!

The trick is unblocking yourself when you get stuck! -> that's what they wanna see

Understand The Main Objective

Repeat: They want to see if you …

- Have passion
- Took the time to learn the basics and practice
- Can think critically
- Can communicate
- Can be trained

What is considered Basic CS Knowledge?

Data Structures	Algorithms	Concepts
Linked Lists	Breadth-First Search	Bit Manipulation
Trees, Tries, & Graphs	Depth-First Search	Memory (Stack vs. Heap)
Stacks & Queues	Binary Search	Recursion
Heaps	Merge Sort	Dynamic Programming
Vectors / ArrayLists	Quick Sort	Big O Time & Space
Hash Tables		

From Ch VII of Cracking the Coding Interview

Where/ How will I code?

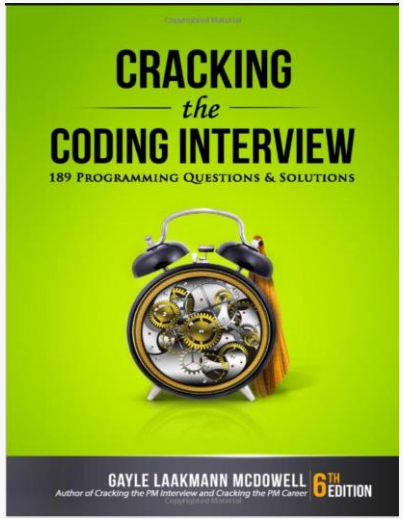
- You choose the programming language
- In online (automated) rounds:
  - You will compile and run your code in an online editor
- With an actual person:
  - plain text editor similar to Google Docs
- No help from the IDE
- Previously this was done on **<u>whiteboards</u>**

- Recreate the environment in your practice

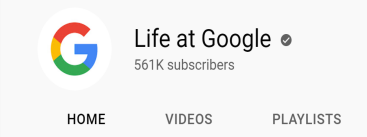
```
#  0 1 2 4 5 6 7 8 9 10 11
# [0 1 1 0 1 1 0 0 1  0  0]
# if I get to index 0, -> there is a path
#[0] -> True
#[1] -> False
# 0 1 2 4 5 6 7 8 9
#[0 1 0 1 1 1 0 0] # minj is 1 maxJ is 3


def dfs(land, index, minJ, maxJ): # 3 4
    print("checking ", index)
    # return True/False if we can reach the start
    if index == 0: # 9  , 4
        return land[0] == 0
    # closer to left of the array
    firstPos = index - maxJ - 1# 4 , 0
    # closer to right
    lastPos = index - minJ - 1 # 5
    if lastPos >= 0:
        firstPos = max(0, firstPos)
        for newIdx in range(firstPos, lastPos + 1): # 4-6,
            if land[newIdx] == 0:
                adjResult = dfs(land, newIdx, minJ, maxJ)
                if adjResult:
                    return True
    if land[index-1] == 0:
        if dfs(land, index-1,minJ, maxJ):
            return True

    land[index] = 1
    return False

def isPath(land, minJ, maxJ):
    if not land:
        return False
    if land[0] == 1:
        return False
    return dfs(land, len(land) -1, minJ, maxJ)
```

Resources for Practicing Technical Interviews

- Your courses: Discrete math, Alg I, CS 251/241
- Cracking the Coding Interview by Gayle McDowell
- Platforms like Leetcode
- A ton of good/bad resources on Youtube, Linkedin, and probably your neighbor's grandma!



Useful Material on Youtube?
if you must (Does not count as practicing)



We are going to talk about how to study and practice next

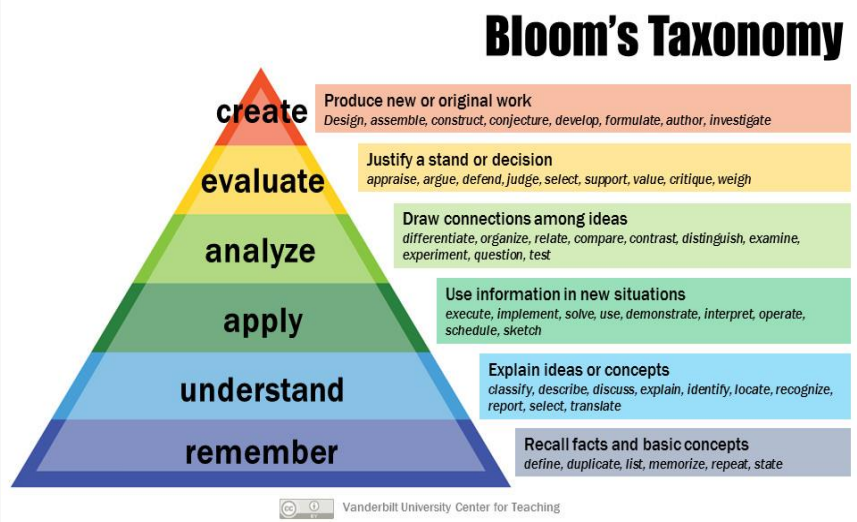
Any burning questions about the resources?

All I Want to Say in the Slides About Practicing

- Learn material at a deeper level (evaluate & analyze)
- Keep track of patterns, themes, your mistakes, …
- Treat practice sessions as the actual interview
- Quality >> Quantity
- It is not fun, but keep practicing!

How to Practice

- Review CS topics (Refer to the "basic CS knowledge slide)
  - CCI / your algorithms/DS course/online resources (GFG) (focus on practical stuff)
  - Think about when/why/how you might use these algorithms & DS, trade offs, etc
  - Get your hands dirty as you review theoretical concepts
- Work on your approach/strategy (chapter VII in CCI)



> Follow ups

} Interview Questions

SCHOOL OF ENGINEERING

Coding Interview Format Overview Revisited! Again!

By using your n-step approach, you will:

- Panic! Get stuck!
- Discuss the problem and solutions (talk as you think) (if with a real person)
- Evaluate tradeoffs, time/space complexity of different solutions
- Get stuck!
- Code!
- Get stuck!
- Test your code! (conceptual, small test case)
- Find errors! Panic! Fix errors
- Follow ups (my favorite)
- Get stuck some more!

Much more effectively!

How to Practice

- Practice problems using your n-step approach
  - Listen, example, BF, optimize, walk through, implement, test (CCI Ch. VII)
  - Only solve problem from valid sources with good solutions
  - Fight the temptation to read solution before 40 mins-1hr (use the hints if you get stuck)
  - Code & test your solution.
  - **Look for common patterns/mistakes/techniques**
    - Have a notebook
  - **Once you have the basics down, try practicing in pairs (mock interviews)**

How to Practice Cont.

- Practice thinking out loud! (You need to do that at the interview)
- Don't underestimate behavioral rounds (review Nada's presentation)
- Review BIG O!

Remember These Practice Problems?

- Given a BST, find all of the arrays that would result in the same tree, if the array elements were to be inserted into the BST one by one (CCI)

- Given a long string and a shorter string, find all of the permutations of the shorter string in the longer one. (very good example for progression from inefficient solutions to the optimal solution: O(S!*L), O(S^2*L), O(S*L), O(S))

  Example: abdcabefbcadcbfgabcd

  target=abcd

Planning your practice

- Ideally you start long before the actual interview
  - After you do some initial review & practice it becomes easier to plan
- Do not wait too long before applying & interviewing
- Quality >> Quantity
- Do not miss opportunities to interview
  - You lose nothing by interviewing
    - ~ 6 months of wait between interviews for full time
- You will not do good until you mess up! So start messing up!

Interview Preparation Grid

Go through each of the projects or components of your resume and ensure that you can talk about them in detail. Filling out a grid like this may help:

Common Questions	Project 1	Project 2	Project 3
Challenges			
Mistakes/Failures			
Enjoyed			
Leadership			
Conflicts			
What You'd Do Differently			

Questions about Preparing and Practicing?

- ✓ My background
- ✓ Working at the big tech companies, is it worth it!?
- ✓ Overview of coding interview format
- ✓ Coding interview effectiveness debate
- ✓ How to prepare
- ➢ At the interview
- ■ Don't we need to land an interview first!?
- ■ Accepting/rejecting offers
- ■ Once you start

Tips for the Actual Interview

- Stick to your n-step approach and make sure you fully understand the question
- Do not jump into coding
- Write good code (**modularize**, **good names,** etc)
- Always start with an example and the brute force solution (do not start coding it though)
    - Break the problem down. What's the slowest/most troublesome part?
- You are solving the problem **with** your interviewer **not for** her.
    - Don't assume! ASK!
    - Discuss trade offs/ different solutions when appropriate

Interview Tips Cont.

- Keep communicating your thought process (you should have practiced this)
- Have questions ready for the interviewer at the end
- The best outcome is leaving your interviewer feeling that you have worked on a problem together, got stuck & unstuck, and **had a good discussion (like you would with a good colleague)**.
- If you've seen the exact problem before … I know it's tempting… -> DO TELL
- Be Nice!

Questions about the day of the interview?

- ✓ My background
- ✓ Working at the big tech companies, is it worth it!?
- ✓ Overview of coding interview format
- ✓ Coding interview effectiveness debate
- ✓ How to prepare
- ✓ At the interview
- ➢ Don't we need to land an interview first!?
- ■ Accepting/rejecting offers
- ■ Once you start

But don't I need to get an interview first !?

You wanna show you're passionate and you are actively doing something about it!

- Use UNM resources (resume, career fair, mailing lists, Handshake, etc)
  - ESS Career Development Resources specialist contact: esscareers@unm.edu
- Linkedin! (There are scams too, be careful)
- Think about building stuff (apps) and put them on your resume, Github, etc
- Pick interesting course projects and do well on them.
- Alternative to work experience: coding heavy courses, projects, indiv.study, etc
- **Usually applications open Aug/Sept/Oct**
- Do not get discouraged!

Getting an Interview Cont.

- Join student service groups of your interest (NSBE, HESO, etc)
- Special programs for freshman/sophomores: Facebook University, Google Bold, etc
- Talk to people! See if you can get referrals
- Application time for interns & new grads: **Aug/Sept/Oct** (for **both** fall & spring grads)

Common Misconceptions

- ✕ You need to have M.S or PhD
- ✕ They only recruit from top schools (Stanford, Cal, MIT, etc)
- ✕ You need to have an IQ of 170 and all A+++s!

When choosing between offers

- Understand your compensation package
  - Salary
  - RSU
  - Bonus
  - Other benefits
- Negotiate
- Feel/create the power to reject companies
- Get as much inside info about the job/environment/group as possible

On The Job

- Create your support network
- Show enthusiasm & take initiative
- Set communication expectations with your manager
- Clarify performance criteria
- NEVER complain about others! Focus on specific problems not individuals!
  - Do not hesitate to escalate through the proper channels if it is an extreme scenario
- Fight the imposter syndrome!
- Be smart, pay attention
- Take advantage of company benefits :)

Thank You!

Time For Questions & Discussion

Topics we covered in Part II

- How to prepare
- At the coding interview
- Landing an interview
- Accepting/rejecting offers
- Once you start

Topics we covered in Part I

- Working at the big tech companies, pros & cons
- Overview of the coding interviews
- Coding interview effectiveness debate