Preparing for Interviews

EECS 281

MAXIM ALEKSA

maximal.io

Agenda

Choosing companies

Recruiting

Résumé

Interview types

Preparing for interviews

Going to an interview

Mock interviews

Disclaimer

Recruiting

Choosing companies

What are you looking for from an internship?

Would you be excited to work on the product?

Would you be excited to work with the people?

Do the things you value align with what the company values?

How big is the company?

Does the company have programs targeted to you?

Recruiting

Apply online

Attend recruiting events

Reach out and meet recruiters

Referral

Recruiting

Online coding challenge

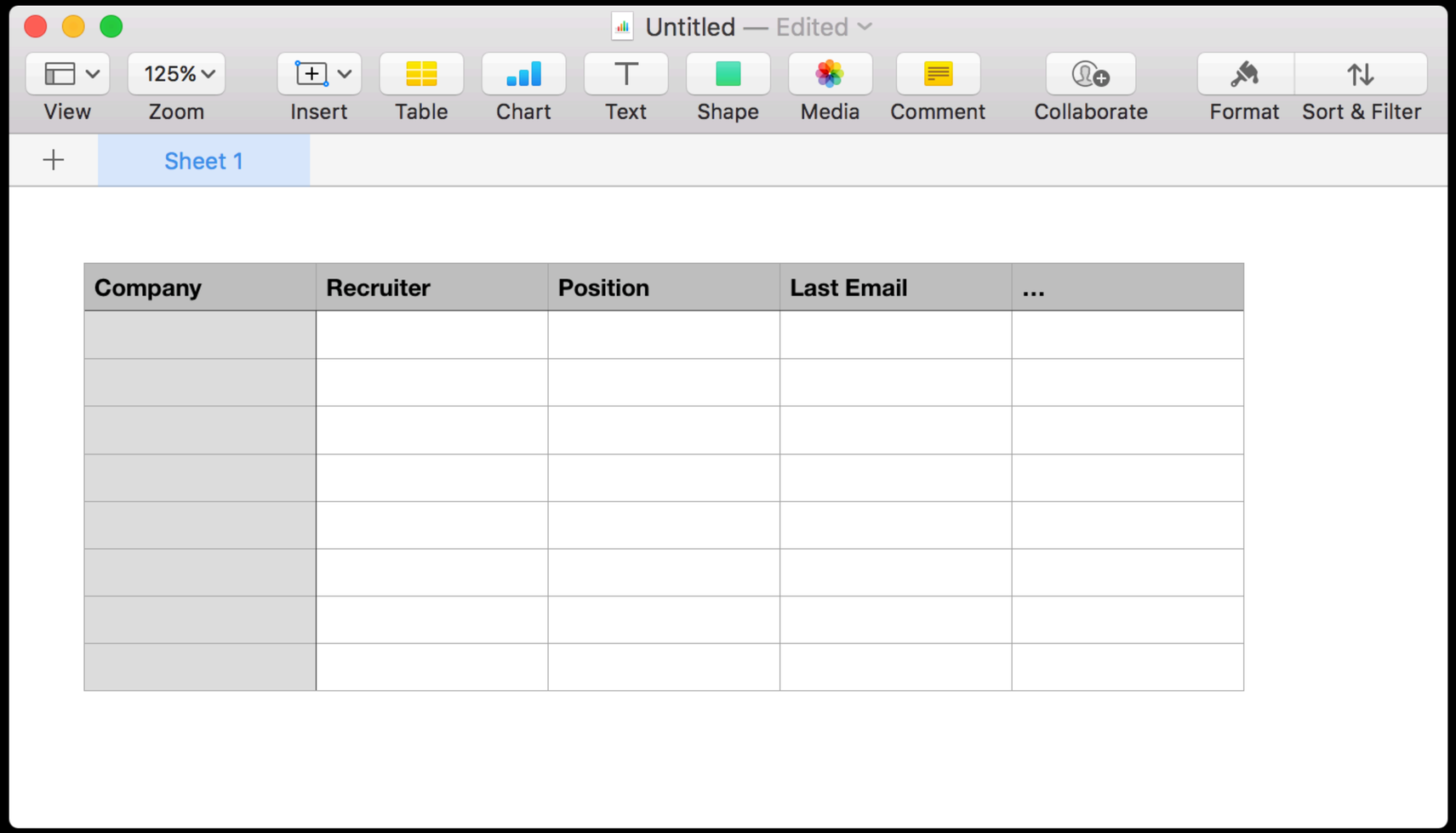
On-campus interview

Phone interviews

On-site portion

Offer stage

Recruiting



Company	Recruiter	Position	Last Email	...

Résumé

30

seconds

Résumé

Recruiters spend very little time looking at each résumé

The first chance to sell yourself

Résumé

One page

Make easy to read and to skim

Make contact information easy to find (and correct!)

Highlight specific accomplishments

Include interesting projects

Don't put charts, ratings or starts

Everyone knows that references are available upon request

Include link to website, professional-looking email address, LinkedIn, GitHub, etc.

Use reverse-chronological ordering

Put only relevant information

Résumé

Critique examples found on Google and on Facebook

Have multiple versions of your résumé

Highlight specific projects and skills

Don't forget to proofread your résumé!

Projects

Everyone goes through the same courses and does the projects

Personal projects make you stand out from others!

Create a personal website and buy a memorable, easy-to-type and easy-to-remember domain name

Have your résumé as an HTML page

Consider adding a downloadable PDF

Be careful with address and phone numbers

Showcase your projects and give them a web presence

Interviews

Why are there interviews on algorithms and coding?

Interview tips

Start interviewing as early as you can (late summer or early fall)

Plan interviewing into your semester schedule

Be aware of time zones

Every interview is a conversation, not a test!

Take control of the interview

Avoid silence

Types of interview

Technical

- Algorithms
- Coding
- System Design
- Knowledge-Based

Non-Technical

- Behavioral
- Culture

Technical interviews

Know algorithms and data structures

Know programming languages, UNIX and terminal, best practices (e.g,. debugging and git) and tools (e.g., Xcode)

Algorithms

Most common

Getting an algorithm right is much more important

Optimize for time and space complexity

OK strategy to start with a simple, naïve solution and then optimize. Much better to have something written than nothing at all!

If the solution seems too complicated, it probably is

Common types of problems

Strings

Recursion

Sorting

Searching

Dynamic Programming

Graphs, Trees

Math

Tools for solving problems

Arrays and linked lists

Hash tables and dictionaries

Built-in string manipulation functions

Memoization for dynamic programming

Algorithms from EECS 281

…

Coding

Implementation could be complex

Decompose the problem

Create helper functions

Make the code simple

Be familiar with the language

Systems Design

Technical design: how would you architect Canvas?

Product design: what feature is missing from Canvas?

Very open-ended!

Understand constraints of the system

State assumption (e.g., “I assume you mean the mobile app”)

Think about the consequences of each decision you make

Challenge your own ideas: when do they break?

Think about future and scalability

Non-technical interviews

Just as important as technical ones!

Could include questions about your experience

Could also be about how well you fit the company

Be prepared to answer questions that might put you under pressure

Culture

How do your values align with the company's values?

How well do you understand the company's values?

Company’s history and division-making process

Your own experiences

Forward-looking

Non-technical tips

Take some time to reflect

Avoid being negative about your past experiences

Don't break the non-disclosure agreements from your past internships, but be very clear about this to your past interviewer

Preparing for interviews

Long-term preparing

Have a plan!

Spend time coding and preparing for interviews

Do more interviews

Do practice interviews with friends

Ask friends for feedback!

Record yourself!

Practice a wide variety of problems

Short-term preparing

Review the basics (including EECS 281 topics)

Review the language of your choice + standard library

Relax!

Get in the zone

Get ready to talk and think out loud

Going into an interview

You can never overdress

Don't be on time. Arrive early!

Be prepared to write code on a computer or on a whiteboard

Don't be afraid to lead the conversation

Use good English. Avoid slang, words such as “like”, “uhmmm”, etc.

Maintain eye contact

Answer the questions

Make yourself memorable (in a good way)

Try to get to know the interviewer and ask what do they care about


the Office

During the interview

Expect at least one coding problem focused on data structures and algorithms

Explain your approach and its correctness

Analyze time and space complexity of your solution

Write clean code (usually in the language of your choice)

Test your code and improve it

During the interview

Make sure you understand the problem

Don’t make any assumptions

When in doubt, ask if you are allowed to assume something

State assumptions

Ask clarifying questions

Tell your interviewer if you’ve seen the problem before!

During the interview

Start by brainstorming approaches to the solution

Thought process is often more important than your output

Draw diagrams or run through test cases

At a high level, explain your chosen approach and reasoning

Convince your interviewer this will work (efficiently)

During the interview

Communicate your progress clearly and out loud

Code quality matters

Over-communicating is better than under-communicating

You and the interviewer are on the same team

Break your code into helper functions and test them

Debugging techniques for coding on a computer and on paper

Pay attention to as many hints as you can!

If you get stuck, start thinking out loud. Avoid long silences!

After the interview

Always have questions prepared at the end of the interview

A way to make sure the company and the role are right

Follow-up with an email and thank for the opportunity!

If an interview does not go well, don't be hard on yourself and don't be discouraged

Think of this as interview practice for next time!

Resources

maximal.io/eecs281

Resources

EECS 281 lecture slides and videos

EECS 281 slides and videos (`http://maximal.io/eecs281`)

`https://datastructures.maximal.io`

Introduction to Algorithms (CLRS) textbook

Resources

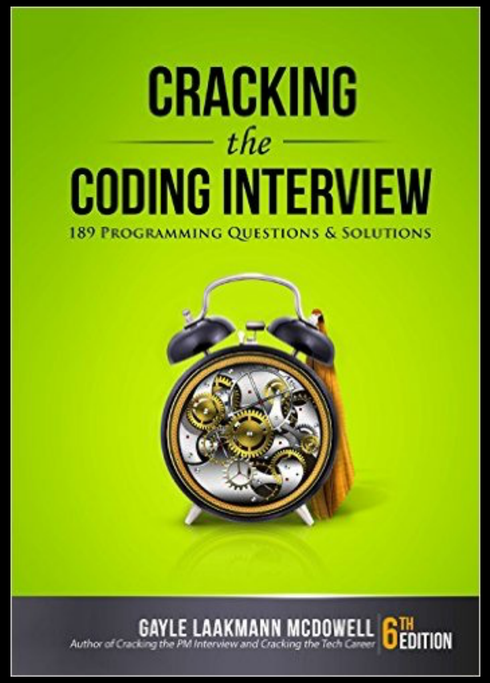
HackerEarth

HackerRank

LeetCode

GeeksforGeeks (Interview Corner)

Resources



Cracking the Coding Interview

Programming Interviews Exposed

Silicon Valley

http://www.hbo.com/silicon-valley

Q&A

Mock interviews