Hack the Internship

Hung Vu
CS'21

Agenda

- Companies
- Interview process
- Tips & tricks
- Q&A

My background

- Summer 2020 

- Fall 2020 

- Spring 2021 
- Summer 2021 

Why you should intern?

- Find out what you want to do for the early part of their career
- Earn extra money
- Work experience to make them more visible for future jobs down the line

BUT...

Internships are **not that** important. They're not the end-all be-all of your career. Most great engineers you know most likely did not do an internship when they were in college.

Companies

Big Tech

- Lots of employees
- Large intern class sizes (1000+)
- Mentorship opportunities
- Great tech tooling
- Diversity initiatives to help recruit URM talent (Google STEP, FBU, AFE)
- Well-developed programs to sponsor employees for green-cards for international students
- WLB varies a lot team-to-team
- Due to large sizes, hiring might be slower
- Located in major cities (SF, NYC, Seattle, Austin, Boston, ...)

“Unicorn” startups/mid-sized tech companies

- Smaller company, fewer employees
- Still has good infra/tooling and mentorships
- “Unicorn” refers to $1bn valuation (usually these companies have upwards of 1000 employees)
- Standard recruiting processes
- Harder to get into due to smaller size
- Mostly in San Francisco Bay Area, Seattle, or NYC

Well-established tech companies

- Strong managers and a history of success
- Bread and butter of the industry, and make up a huge proportion of the internship engineering pool
- Usually have strong WLB
- Legacy codebases
- Spread across the US (e.g Chicago/Tempe/Boston/NYC/Westlake/Palo Alto/Austin/Clearwater etc)

Early-stage startups

- IMPACT
- Entrepreneurship opportunities
- Less mentorship
- High variance
- Due to the nature of early-stage startups, you may have to work much longer hours
- Sometimes do not recruit through regular means – you may have to reach out through email or in-person
- Hire for interns that can work across the stack
- Many in San Francisco Bay Area, but can be found all across the US

SEQUOIA   Combinator greylock

Non-tech companies

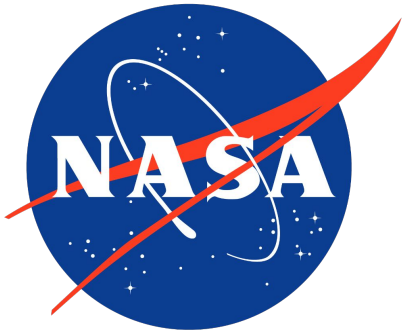
- Great WLB
- A specific industry (such as Commerce or Transportation)
- Rapidly modernizing
- Some are known for their strong engineering departments, while others are not. Do your research!
- Located in cities where many tech companies aren't located
- Fun fact: If you intern at Southwest, you get unlimited free flights during your internship!

Government contractors

- Cybersecurity or Defense technologies
- Known for focusing more on human/behavioral aspect of interviews
- Strong work-life balance and amazing benefits
- Tricky rules and regulations (e.g filling paperwork to visit certain countries, etc)
- Being a US citizen or national is often a requirement for working on many federal contracts
- Mostly based in DMV

NORTHROP GRUMMAN

Raytheon Technologies



Banks

- Quickly modernizing financial industry
- Finance
- Some are known for asking case studies questions, but this can vary
- Many banks, such as Capital One, are making a conscious effort to be a technology company first, and a bank second.
- Mostly based in NYC

Hedge funds/trading firms

- Build software to trade stocks/options
- 3 types of CS roles: engineering, trading, and quant research
- Ask difficult brainteasers and math/systems questions
- Pay a lot
- WLB can be great at some companies (9-5, trading hours) and really long at others (60-80 hrs/w). Do your research!
- Mostly based in Chicago/NYC

Timeline

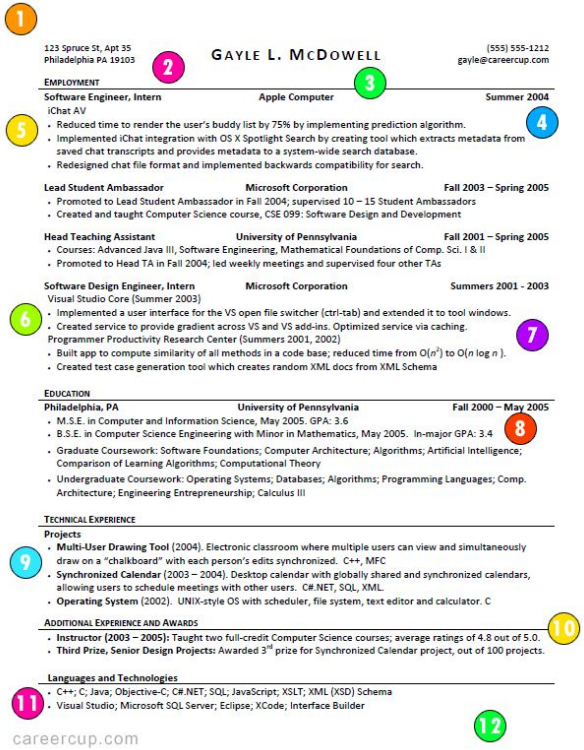
- Larger tech companies: August-January
- Smaller tech companies: August-May
- Length:
  - typically 12 weeks
  - 10-16 weeks
  - Fall/Spring internship can vary.

Interview Process

- **Resume screen**
- Coding challenge
- Recruiter phone screen
- **Technical phone interview**
- **(Virtual) Onsite interview**
- Behavioral/team match/hiring manager interview

Resume screen

- Your resume will be submitted to a pool of thousands applicants
- Review:
  - Automatically by machine
  - Manually by recruiter
- Next steps: recruiters reach out for coding challenge/phone call
- Ghosted
- Rejected



123 Spruce St, Apt 35
Philadelphia PA 19103

GAYLE L. MCDOWELL

(555) 555-1212
gayle@careercup.com

EMPLOYMENT

Software Engineer, Intern — **Apple Computer** — **Summer 2004**
iChat AV
- Reduced time to render the user's buddy list by 75% by implementing prediction algorithm.
- Implemented iChat integration with OS X Spotlight Search by creating tool which extracts metadata from saved chat transcripts and provides metadata to a system-wide search database.
- Redesigned chat file format and implemented backwards compatibility for search.

Lead Student Ambassador — **Microsoft Corporation** — **Fall 2003 – Spring 2005**
- Promoted to Lead Student Ambassador in Fall 2004; supervised 10 – 15 Student Ambassadors
- Created and taught Computer Science course, CSE 099: Software Design and Development

Head Teaching Assistant — **University of Pennsylvania** — **Fall 2001 – Spring 2005**
- Courses: Advanced Java III, Software Engineering, Mathematical Foundations of Comp. Sci. I & II
- Promoted to Head TA in Fall 2004; led weekly meetings and supervised four other TAs

Software Design Engineer, Intern — **Microsoft Corporation** — **Summers 2001 - 2003**
Visual Studio Core (Summer 2003)
- Implemented a user interface for the VS open file switcher (ctrl-tab) and extended it to tool windows.
- Created service to provide gradient across VS and VS add-ins. Optimized service via caching.

Programmer Productivity Research Center (Summers 2001, 2002)
- Built app to compute similarity of all methods in a code base; reduced time from $O(n^2)$ to $O(n \log n)$.
- Created test case generation tool which creates random XML docs from XML Schema

EDUCATION

Philadelphia, PA — **University of Pennsylvania** — **Fall 2000 – May 2005**
- M.S.E. in Computer and Information Science, May 2005. GPA: 3.6
- B.S.E. in Computer Science Engineering with Minor in Mathematics, May 2005. In-major GPA: 3.4
- Graduate Coursework: Software Foundations; Computer Architecture; Algorithms; Artificial Intelligence; Comparison of Learning Algorithms; Computational Theory
- Undergraduate Coursework: Operating Systems; Databases; Algorithms; Programming Languages; Comp. Architecture; Engineering Entrepreneurship; Calculus III

TECHNICAL EXPERIENCE

Projects
- **Multi-User Drawing Tool** (2004). Electronic classroom where multiple users can view and simultaneously draw on a "chalkboard" with each person's edits synchronized. C++, MFC
- **Synchronized Calendar** (2003 – 2004). Desktop calendar with globally shared and synchronized calendars, allowing users to schedule meetings with other users. C#.NET, SQL, XML.
- **Operating System** (2002). UNIX-style OS with scheduler, file system, text editor and calculator. C

ADDITIONAL EXPERIENCE AND AWARDS
- **Instructor (2003 – 2005):** Taught two full-credit Computer Science courses; average ratings of 4.8 out of 5.0.
- **Third Prize, Senior Design Projects:** Awarded 3rd prize for Synchronized Calendar project, out of 100 projects.

Languages and Technologies
- C++; C; Java; Objective-C; C#.NET; SQL; JavaScript; XSLT; XML (XSD) Schema
- Visual Studio; Microsoft SQL Server; Eclipse; XCode; Interface Builder

careercup.com

Coding challenge

- Initial coding challenge to assess your coding skills
- Some are automated, some are selected
- 1 week before expires. Can vary from 48 hours to 2 weeks
- 1-3 hour(s) to complete by yourself
- Leetcode/API questions
- Popular platforms:
  - HackerRank
  - CodeSignal
  - Codility
  - Take home assignments

Recruiter phone screen

- 15-30 mins
- Work/project experience
- Why you are interested in that company?
- Interview process
- Your timeline/expiring offers

Technical phone interview

- Video/audio/phone call
- Share screen/Collaborative live coding session (Coderpad)
- 1-2 SWE(s) will call you to start the interview
- First 10 mins: Introduction, work experience, quick concept questions
- Next 30-45 mins: Coding question(s)
  - How to reverse a linked list
  - Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target. (TwoSum on Leetcode)
- Last 5-10mins: Time for questions about the engineers/company
  - When you were looking for a job last time, what made you choose this company?
  - What are your expectations of the person who takes this role? What would define success for this role?

Technical phone interview

- Tell the interviewer your thoughts on how to solve the question
- As you write your solution, talk about your code
- Periodically write tests at the bottom of the file to make sure your functions are working as intended (these don’t need to be fancy or in their own method e.g you can just use assert statements).

Onsite interview

- Some tech companies will fly you out for an interview at their HQ. Often, they'll take care of plane ticket costs, food reimbursements, housing, etc.
- The interview will be composed of 3-4 technical interviews (e.g solving algorithm questions on a whiteboard or writing code on a computer), and sometimes a behavioral interview with a manager.
- Sometimes companies will give remote final round interviews (through the phone or internet). This is often the same format as a technical phone interview, just longer.

Behavioral interview

- Usually the last round of the process. This means that your technical skills match their bars.
- Vibe check
- Team matching
- Non-technical questions
    - Tell me a time when you work in a group.
    - Tell me a time when you failed.
    - Your biggest achievement
- Use STAR method (Situation, Task, Action, Result)

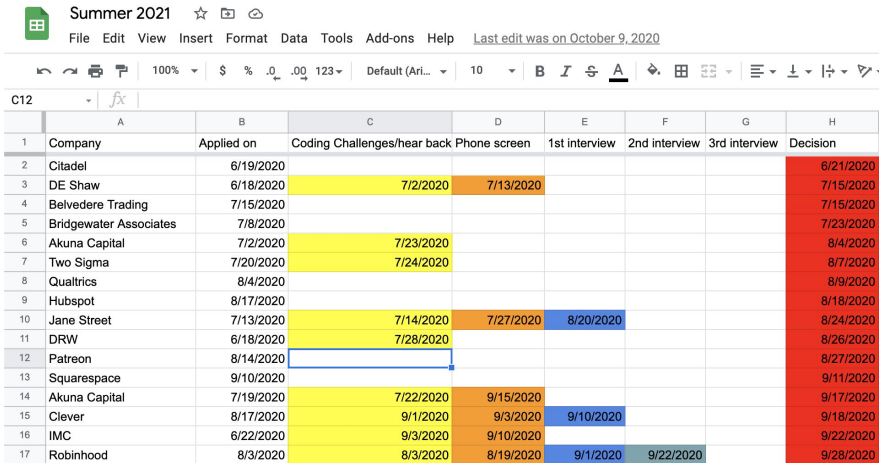
Fixing your resume

Recruiters read through thousands of resumes a week, so make sure yours is in tip-top shape

- Save it as a PDF, name it “Your Name - Resume - Year”
- One page only
- Be sure to include tech stack, impact, specific detail of what you did rather than just “I implemented a feature.”
- Use a template: https://www.careercup.com/resume or LaTEX
- Make sure resume is parseable.

Applying to Companies

- Apply to as many companies as you can
- Make a Google spreadsheet to track your progress
- If you know someone at a certain company, ask them for a referral! This is the best way to ensure a recruiter will reach out to you and start the interview process.



Summer 2021

File Edit View Insert Format Data Tools Add-ons Help Last edit was on October 9, 2020

C12

	A	B	C	D	E	F	G	H
1	Company	Applied on	Coding Challenges/hear back	Phone screen	1st interview	2nd interview	3rd interview	Decision
2	Citadel	6/19/2020						6/21/2020
3	DE Shaw	6/18/2020	7/2/2020	7/13/2020				7/15/2020
4	Belvedere Trading	7/15/2020						7/15/2020
5	Bridgewater Associates	7/8/2020						7/23/2020
6	Akuna Capital	7/2/2020	7/23/2020					8/4/2020
7	Two Sigma	7/20/2020	7/24/2020					8/7/2020
8	Qualtrics	8/4/2020						8/9/2020
9	Hubspot	8/17/2020						8/18/2020
10	Jane Street	7/13/2020	7/14/2020	7/27/2020	8/20/2020			8/24/2020
11	DRW	6/18/2020	7/28/2020					8/26/2020
12	Patreon	8/14/2020						8/27/2020
13	Squarespace	9/10/2020						9/11/2020
14	Akuna Capital	7/19/2020	7/22/2020	9/15/2020				9/17/2020
15	Clever	8/17/2020	9/1/2020	9/3/2020	9/10/2020			9/18/2020
16	IMC	6/22/2020	9/3/2020	9/10/2020				9/22/2020
17	Robinhood	8/3/2020	8/3/2020	8/19/2020	9/1/2020	9/22/2020		9/28/2020

Get noticed

- Get a referral from an employee or former intern
- Talk to recruiters at career fairs and hand in your resumes
- Go to company events/tech talks and check-in/hand in your resumes
- Email recruiters directly by asking them for their card in-person
- Find recruiter emails and cold email them

Preparing for your interviews

- The vast majority of companies now ask algorithm questions during their interviews. The best way to get an internship is to get better at answering these algorithm questions.
- Know your preferred language well
- Brush up your Data Structures and Algorithms knowledge

Preparing for your interviews

- Cracking the Coding Interview (CTCI)
  - A book on the most common algorithms/data structures from interviews
- Leetcode
  - Online coding platform where you can find and practice common algorithms/data structures problems.
  - Easy/medium/hard
- *Elements of Programming Interview*
  - *Similar to CTCI, except significantly harder.*
- *Codeforces/Competitive Programming/ICPC*
  - *Competitive programming platform where you can compete with others to solve algorithmic questions in the fastest time possible.*

How to practice technical problems

- Write out your solutions on a blank sheet of paper first
- If you can't come up with an idea of the solution in 15 minutes, then look up the solution (look in the back of the chapter for Leetcode, in the Discuss tab for Leetcode, or Google it) and retry
- While you're thinking, talk through your thought process (it doesn't come natural, so practice!)
- When you have a solution, code it up on your computer and test to see if it works (try to get it working on the first try, if possible)
- Mock interviews
  - https://www.pramp.com/
  - https://interviewing.io/
  - https://triplebyte.com/
  - https://alist.co/

Resources

- Reddit
  - r/csMajors
  - r/cscareerquestion
- Discord
  - CS Majors (https://discord.gg/mW9py9e)
  - cscareers.dev (https://discord.com/invite/cscareers)
- LinkedIn
- GitHub
  - https://github.com/pittcsc/Summer2022-Internships
- Salaries:
  - https://www.levels.fyi/internships/

But...

- This process is extremely overwhelming
- Please take some rest
- It’s ok to fail
- No matter how much you prepare, you will fail a proportion of your interviews, for reasons outside of your control.
- 150, 10, 3, 2

Summary

- Fix your resume
- Create a jobs spreadsheet and start applying to internships
- Practice technical interviewing
- Work on problems on a piece of paper, vocalize your process, and code up your solutions
- Find a copy of CTCI and do the problems from Chapters 1-4
- After that, practice on Leetcode and schedule some mock interviews

Q&A