

EMILY HUYNH

INTRO TO TECH INTERVIEWS

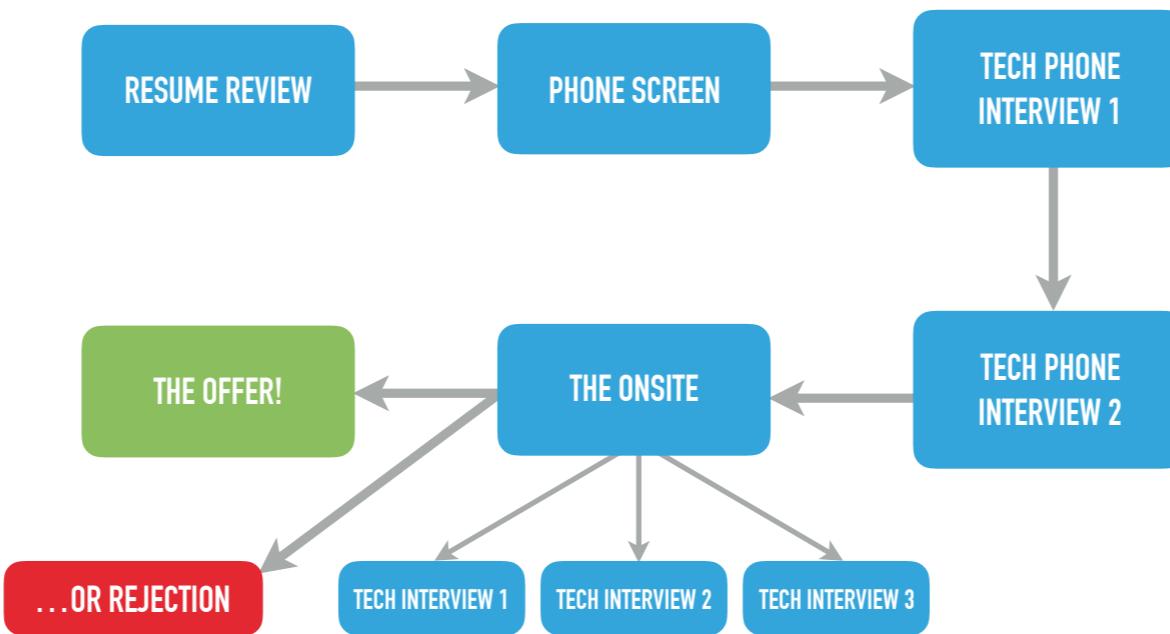
This presentation is specifically for software engineering roles.



HI, I'M EMILY

CS Major on STEM Path to MBA
Brother of AKPsi since Spring 2014
Web & Design Director of Crimson Kindness

INTERVIEW WORKFLOW: EXAMPLE



This is a typical interview process for internships and full-time jobs. Keep in mind, most companies customize this by adding/removing steps. Some companies may skip the phone screen, sometimes you don't have an onsite, etc. Additionally (and unfortunately), you can be rejected at any point in the process.

Phone interviews => coding in a shared document (or Google Doc if you're interviewing with Google)

Onsite => pen & paper, whiteboard, or computer



WHAT IS A TECH INTERVIEW?

- ▶ 45-60 minute interview with an engineer
- ▶ 1-2 coding questions
- ▶ A way to gauge your technical ability
- ▶ Complete BS

An individual tech interview is a 45-60 minute interview with an engineer.

If you can't solve the problems within the allotted time, that doesn't mean you're a bad programmer or engineer. It just means you haven't practiced enough tech problems. The most important thing to remember is that your performance in a technical interview is not an indicator of your intelligence - just like standardized tests.

3 COMMON TYPES OF TECHNICAL QUESTIONS

1. High level/project-based questions

- ▶ Tell me about a project you've worked on.
- ▶ What is polymorphism?

2. Design scenarios

- ▶ How would you design/implement Venmo?

3. Coding problems

- ▶ What we'll focus on today

- 1) Hackathon projects. They want to know what projects you've worked on and what you've learned in class. Can anyone answer this?
 - 1) **Polymorphism** allows each subclass has a way of implementing a function
 - i.e. Animal class: Bird, Turtle implements move() differently
- 2) What you've learned in school and your general thought process
 - 1) Was asked this question in one of my interviews. She really wanted to know my thought process and ensure that I was familiar with object oriented design.
Can anyone answer this?
 - User : send, invoice
 - Transaction: amount

CODING INTERVIEWS EVALUATE:

- ▶ your knowledge of data structures
- ▶ your ability to develop algorithms
- ▶ your teamwork ability
- ▶ your ability to solve problems



- You **need** to know your data structures. Can anyone name a few?
 - Arrays, Lists, Queues, DICTIONARIES (aka HashMap)
- Most coding questions ask you to develop an algorithm to solve a problem. Most of you have this experience in your programming projects, especially if you're in/have taken CS 200/360: Data Structures & Algorithms.
 - An algorithm is a set of instructions to solve a problem.
- Your interviewer is there to help you. If they give you advice, you should take it. Basically, your interviewer will use this time to evaluate whether or not you'd be able to work with engineering teams. The main takeaway is to be responsive to their advice and to ask questions. **It's very important to talk through your thought process!**
- Finally, your interviewer will expect you to solve the problem in the allotted time. There are some instances where you don't have to solve it completely, but those are rare.

STRING PARSING EXAMPLE: DUPLICATES()

- ▶ Determine if a string of standard ASCII characters contains duplicate characters
- ▶ Input: “abc123”
- ▶ Output: **false**
- ▶ Input: “abc12a3”
- ▶ Output: **true**

Another example of a common string-parsing problem is the palindrome problem, where you have to determine whether or not the input string is a palindrome.

THE SOLUTION

```
def has_duplicates(string):  
  
    # because there are only 128 standard ASCII characters  
    if len(string) > 128:  
        return True  
  
    chars = set()  
  
    for c in string:  
        if c not in chars:  
            chars.add(c)  
        else:  
            return True  
  
    # if we've made it here, then no duplicate chars  
    return False
```

HARDER EXAMPLE: BEST TIME TO BUY AND SELL STOCK

- ▶ Say you have an array for which the i th element is the price of a given stock on day i .
- ▶ If you were only permitted to complete at most one transaction (ie, buy one and sell one share of the stock), design an algorithm to find the maximum profit.
- ▶ Input: [7,1,5,3,6,4]
- ▶ Output: 5
- ▶ Input: [1,2,3,4]
- ▶ Output: 0

Palantir Interview Question - <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

THE SOLUTION

```
def maxProfit(prices):
    # if the array is empty
    if len(prices) == 0:
        return 0

    # declare our variables
    min_price = prices[0]
    max_profit = 0

    for p in prices:
        # check if we have a new max profit
        if p - min_price > max_profit:
            max_profit = p - min_price

        # check if we have a new min profit
        if p < min_price:
            min_price = p

    return max_profit
```

Palantir Interview Question - <https://leetcode.com/problems/best-time-to-buy-and-sell-stock/>

NEXT STEPS:

- ▶ Cracking the Coding Interview
 - ▶ 6th edition
- ▶ LeetCode
- ▶ HackerRank
- ▶ CareerCup
- ▶ Practice, practice, practice

apply to CrimsonHacks

CtCI: 6th edition has hints, helps to simulate actual interviews. Provides a fantastic overview of the basics, but you shouldn't use this as your sole resource!

LeetCode: Google pulls interview questions from here! Start with easy questions and then work your way up. The buy/sell stock problem is from here.

HackerRank: Easier problems, interview questions also pulled from here

CareerCup: Actual interview problems from companies like Google, Epic, Pinterest, etc.

The more consistently you practice, the better you will be at tech interviews.

Also - apply to CrimsonHacks! Hackathons are a great way to get projects on your resume. You can do so at <http://CrimsonHacks.com>.

THANKS!

QUESTIONS?

slides + notes available at bit.ly/akpsi-tech

contact me via slack, email, or facebook