

# How to prepare tech interviews

Jieung Kim

Aug. 2022

# Contents

# Contents

- Tech interview overview
  - What is tech interview?
  - Interview process and format
  - What do we need to do? – in general
- Coding interview
  - Coding interview process
  - What do we need to do? – specific for coding interview
  - Coding interview example - step by step
  - Coding interview example (videos)
  - Feedback format

# Tech Interview Overview

# What is tech interview?

- It's a specialized, rigorous process that **tests your coding skills, problem-solving abilities, your skill fit and personality**
- It involves **challenges** and **assignments**
  - They're more like an exam than a typical question-and-answer interview
  - You have to *prove* that you have the skills required to do the job, rather than just tell the interviewer that you have them
  - It's to see how you tackle real-world problems—like the ones you might be facing once you have the job!
- Common misunderstanding: the purpose of the technical interview is to trick you with **brain teasers** or **impossible questions**.  
→ **NO!!!!!!**
  - Some hedge-fund companies still ask those questions

# Kinds of tech interview

- **Depending on topics**

- **Tech phone screening**

- It is conducted right after your resume is accepted
    - This interview stage is designed to see the following things
      - If you are a good fit for the company's need
      - If you are qualified and enthusiastic enough to proceed to the next stage

- **Behavioral**

- It is usually a part of main interviews
    - It checks whether you have a good fit with the company's culture

# Kinds of tech interview

- **Depending on topics**
  - **Algorithm & data structure**
    - It mainly asks for knowledge about algorithms and coding
    - Most big tech firms conduct more than one algorithm & data structure interview for all software engineer candidates
  - **System design**
    - It asks about the overall design for the system
    - This interview is usually not for the entry level
  - **Domain specific interview**
    - It asks about basic knowledge on specific domains
    - This interview is usually not for the entry level
  - **Project interview**
    - It asks about the projects and what you did
    - It could be parts of other interviews (the first 5-10 minutes of other interviews) or the one entire interview

Many people treat the above four as actual technical interviews

# Kinds of tech interview

- **Depending on topics**

- **Pop quiz**

- A quick and dirty way of weeding out extremely weak (or even non-tech) candidates
    - Example
      - What is 4 & 5 (in binary)? Answer: 4
      - What is the time complexity of bubble sort? Answer:  $O(n^2)$

- **Take home assignment**

- To overcome limitations and drawbacks of real-time coding interviews
    - This interview format takes up more time from both the candidates and the company
    - Some startups use this format, but big tech firms usually do not use this
    - Example
      - Build a flights listing app
      - Build a snake game

These two are not as common as the previous ones



# Kinds of tech interview

- **Depending on methods**

- **Phone (virtual) interview**

- Technical phone screening or phone coding interview
    - Phone coding interview
      - It tests your coding skills with a preliminary test before having you actually come in
      - It may be conducted over the phone, via Skype or Zoom, or could be a homework-type assignment

- **Onsite (virtual) interview**

- Many people think this phase is the actual tech interview
    - It involves an in-person interview with coding challenges you have to complete on a whiteboard in front of the interviewer(s)
    - Due to COVID-19, it is mostly performed via virtual

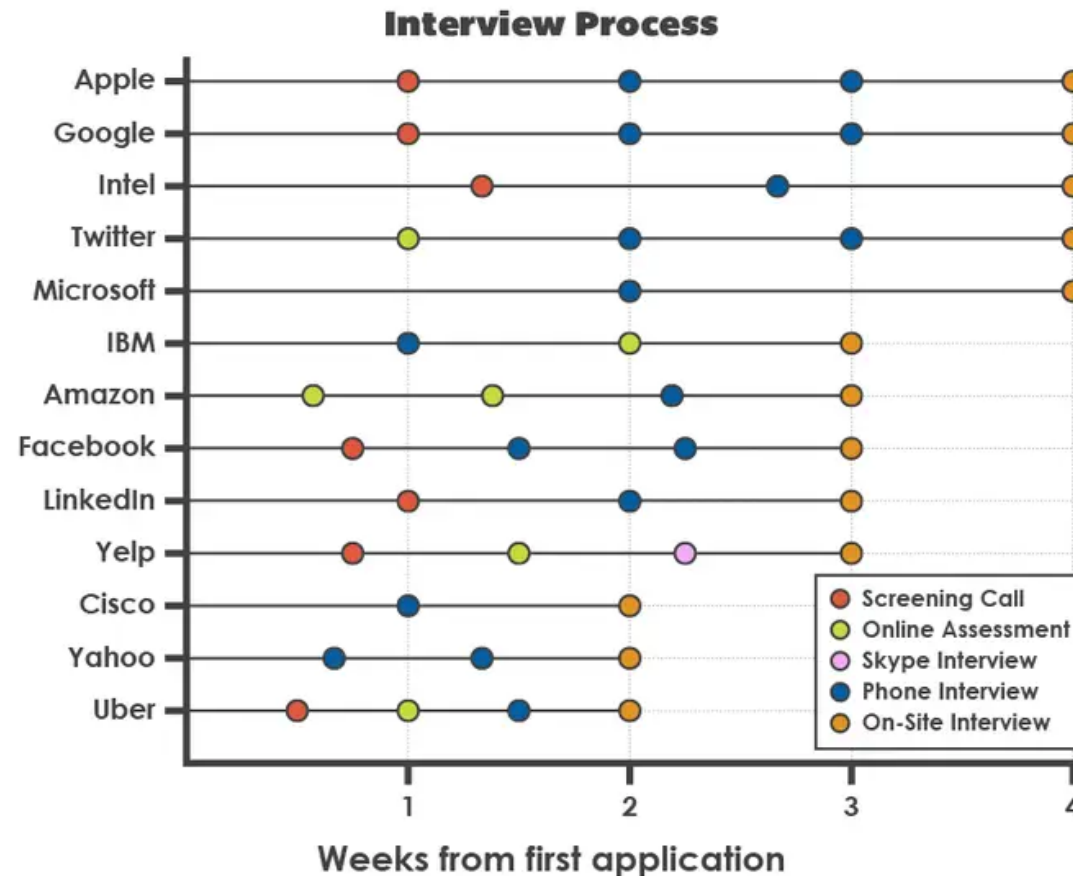
# Interview process and format

- Typical tech recruiting process

Step 1	Express initial interest <i>Career fairs, events, or via email</i>
Step 2	First round of the technical interview <i>Coding challenge, 30 - 60 minute tech screen</i>
Step 3	Final round of interviews <i>In person, 3 - 6 interview rounds lasting ~60 minutes each</i>
Step 4	Decision & offer <i>Good luck!</i>

# Interview process and format

- Different companies have different difficulties and processes



# Interview formats of famous companies

We will look at interview formats of the following companies

- Google
- Facebook
- Dropbox
- Airbnb

Note that these formats can be changed by companies and several roles may have role-specific sessions

# Google interview format

- Google
  - Recruiter phone screen
  - Technical phone interview:
    - 1 or 2 x algorithm on Google Doc
  - On-site (Usually 5 interviews):
    - 1 or 2 x System design or domain specific coding
    - 2 to 4 x Algorithm on whiteboard
    - 1 x General Cognitive Ability, Leadership and "Googleness"
  - Team matching
    - Speak with managers from different teams who are interested in your profile
  - Tips:
    - In rare cases, candidates may even be allowed to skip the phone interview round and advanced to on-site directly
    - Sometimes, you only receive an offer if you are successfully matched with a team



# Facebook interview format

- Facebook
  - Recruiter phone screen
  - Tech phone interviews:
    - 1 or 2 x Algorithm on Skype/CoderPad
  - On-site:
    - 2 x Technical coding interview on whiteboard (Ninja)
    - 1 x Behavioral (Jedi). Meet with an Engineering Manager and discussing past experiences and working style
    - 1 x Design/architecture on whiteboard (Pirate)
  - Tips:
    - For the Jedi round, you may be asked a tech question at the end of it
    - For the Ninja rounds, you may be asked one to two questions depending on how fast you progress through the question



# Dropbox interview format

- Dropbox
  - Recruiter phone screen
  - Technical phone interviews:
    - 2 x Algorithm on CoderPad
  - On-site:
    - 2 x System design or domain specific coding
    - 1 x General coding on CoderPad
    - 1 x All around. Meet with an Engineering Manager and discussing past experiences and working style
  - Tips:
    - You can code on your own laptop and look up APIs
    - Dropbox recruiters are very nice and will give you helpful information on what kind of questions to expect for the upcoming sessions



# Airbnb interview format

- AirBnB
  - Recruiter phone screen
  - Tech phone interview:
    - 1 or 2 x Algorithm on CoderPad
  - On-site:
    - 2 x Algorithm coding on CoderPad
    - 1 x System design/architecture/domain specific coding
    - 1 x Past experience/project
    - 2 x Cross functional





# Airbnb interview format

- AirBnB

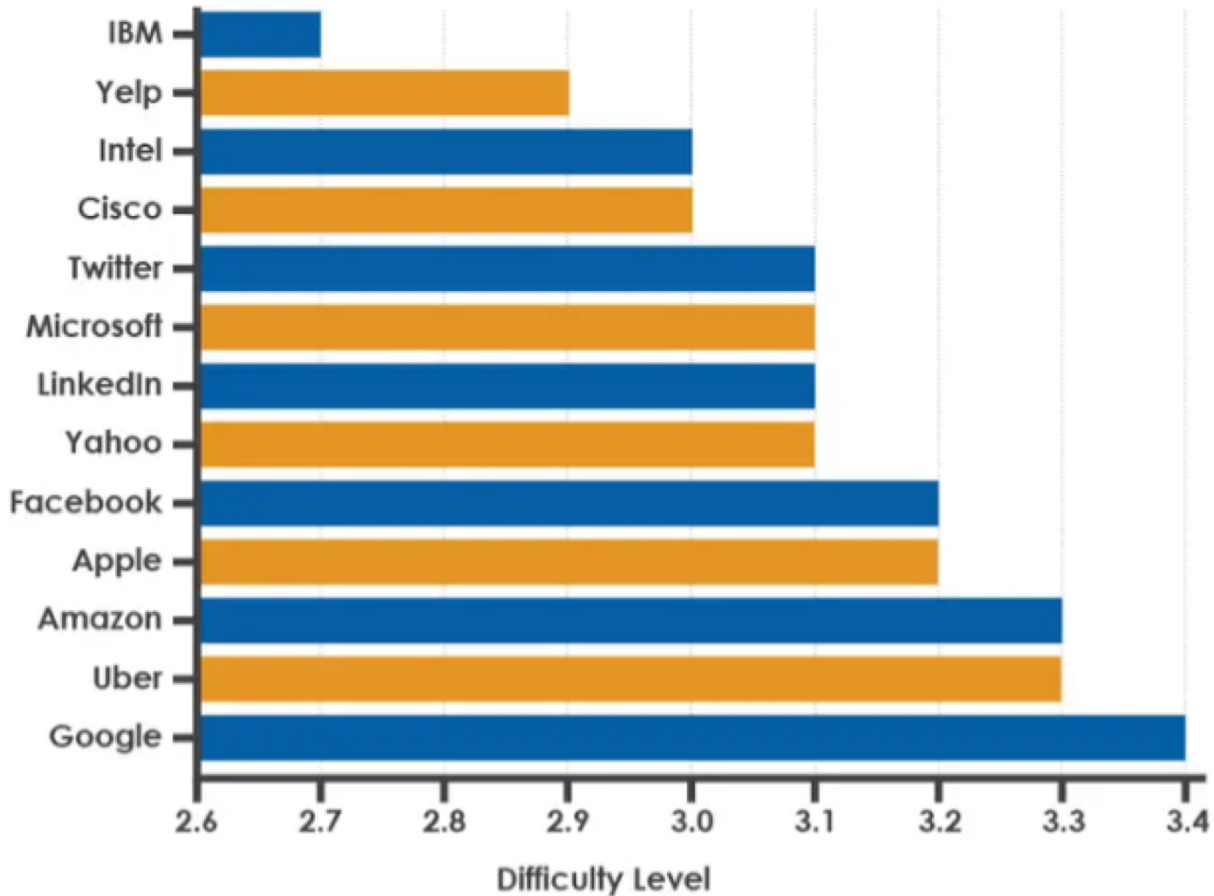
- Tips

- All sessions involve coding on your own laptop
    - Prepare your development environment in advance
    - You are allowed to look up APIs if you need to
    - They seem to place high emphasis on compilable, runnable code in all their coding rounds
    - Cross functional interviews will involve getting Airbnb employees from any discipline to speak with you
    - These interviews are mostly non-tech but are extremely important to Airbnb because they place a high emphasis on cultural fit

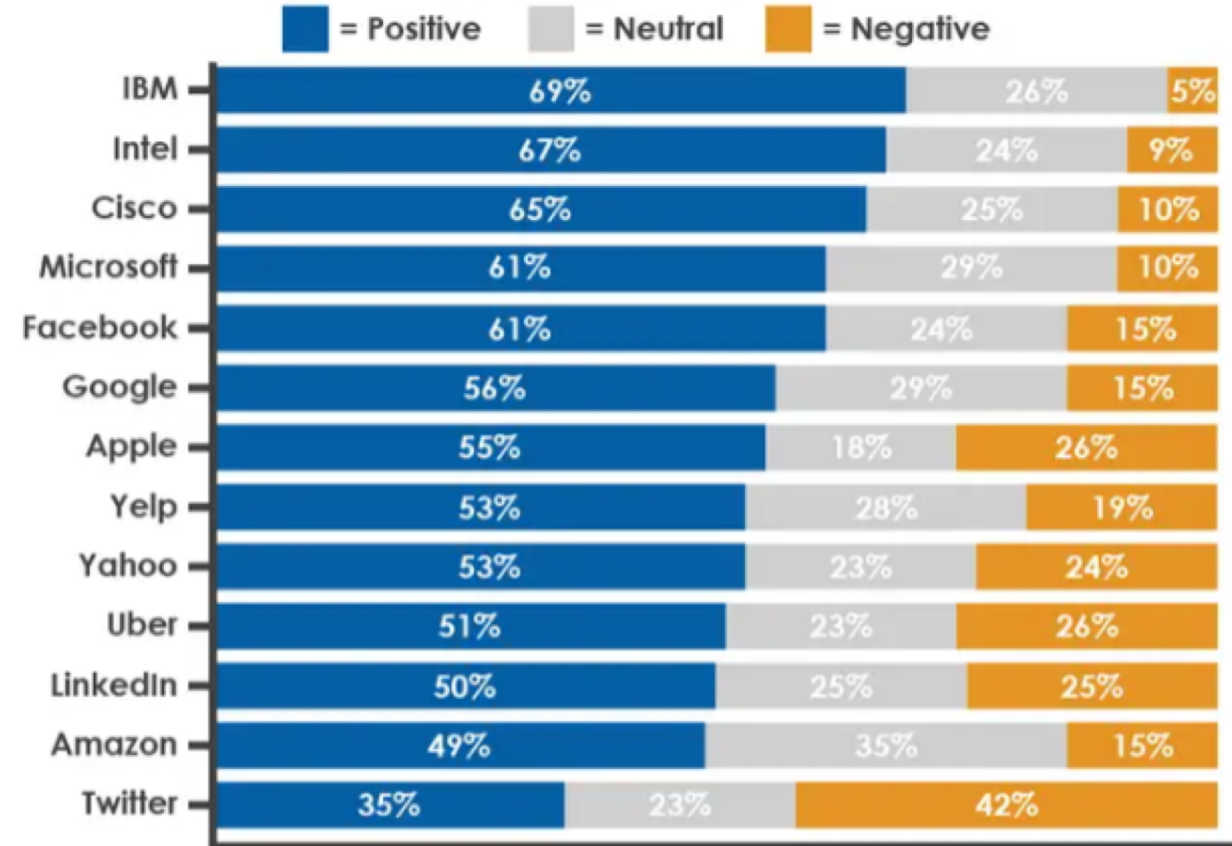


# Interview process and format

**Interview Difficulty**



**Interview Experience**



# What do we need to do? – in general

Focusing on

- Preparation & research
- Practice
- Common pitfalls

Prerequisites

- We expect you have a good resume and skill sets for the company
- Please look at "[Create Your Resume for Google: Tips and Advice](#)" and other sources (at the "[Life at Google](#)" channel) for more details

# Prepare & research

- **Learn about the company:**
  - Visit website
  - Talk to recruiters when at career fairs
  - Visit online job sites
- **Learn about the position for which you are applying:**
  - Ask for a job description before your interview
  - Ask about the position to people at the same company
  - Make a resume that is suitable for the job description

# Prepare & research

- **Research interviewing style:**
  - Talk to people who have been through the interviewing process before
  - Talk to recruiters about what you can expect from your interview day (They will actually provide you interview preparation materials)
- **Prepare interviewers' questions as well as your questions**
  - Make sure you know the things you claim to know
  - Be prepared to ask meaningful questions

# Practice

- **Review your coursework** to make sure you are on top of the material and can effectively discuss concepts
  - Data structure and algorithm are must to do
  - Architecture, network, database, security etc. are important for role-specific interviews
    - E.g., front-end, system design, testing
- **Collect sample interview questions**
  - Google it!
  - Don't forget interviews except algorithm/data structure interviews
    - E.g., leadership, project, architecture, data analysis, etc. based on your applied role
- **Prep with real people**
  - E.g., friends, web-based tools, people at the company, etc

# Common pitfalls

- **No up-to-date resume**
- **Lack of knowledge about job descriptions**
  - The role itself
  - The team that you hope to apply
- **During interviews**
  - Talk without think (take a break!)
  - Assuming interviewer's thought (ask clarification questions)
  - Skip answering (Answer all questions unless interviewers cut you off)
- Etc

# Coding interview



# Coding interview process

The coding interview is the most common, but a scary interview session for many people

It usually **consists of the following steps**

- **Introductions**
- **Project discussion**
- **Coding exercise**
- **Your questions**

Interview format may vary depending on companies/interviewers

# Introduction

- **Develop your pitch**

- Who are you?
- What are your interests? Goals?
- Why are you interested in the position?
- Very short (around 1 minutes).

- Show that **you know SOMETHING about the company** and **why you're interviewing with them**

# Project discussion

- **Pick 1-2 projects off your resume** you can speak in depth about
  - Pick your biggest or most technically interesting project
  - It's ok to talk about school projects
- Don't assume subject domain expertise, but **be able to go in to detail when asked**
- Always **clarify your role** and teammates' roles in those projects
- In several cases, it will be **replaced with warm-up questions**

# Coding exercise

Structure your thoughts

- **Steps 1 & 2. Repeat the Problem and Give Examples**

- clarify the question
- talk through sample input and expected output

- **Step 3. Approach**

- **Brute force:** what is the simplest way to solve this?
- **Optimize:** can you save run time or memory?
- **Walk through:** clarify your algorithm with complexity

- **Step 4. Code**

- write the code

- **Step 5. Test and Big O**

- list test cases, walk through your code with test cases, and calculate time (and space) complexity

# Your questions

Interviewers will mention your questions will not be part of assessment, but...

- **Show your interest! with good questions.**
  - "What is your favorite part about working for X?"
  - "What are some projects you've worked on at X?"
  - "Where do you see X in 5 years?"
- **Don't ask rude questions.**
  - "Did I pass?"
  - "How much do you make?"
  - "What is your salary?"

# What do we need to do? – specific for coding interview

- Decide on a programming language
- Study CS fundamentals
- Internalize the Do's and Don'ts of interviews
- Practice solving algorithm questions & doing mock interviews

# Decide on programming languages

- **Strongly recommended: Pick something OOP**
  - C++, Java, Python
  - **Python** is strongly recommended thanks to its rich libraries
  - But **pick up your favorite language** is better than using unfamiliar OOP
    - When choosing C, you can assume some basic data structures and functions on it (e.g., sort, hashtable, etc.) are pre-defined
    - When you assume them, please ask about them to the interviewer whether it's OK
  - Some roles require specific languages (e.g., Objective-C/Swift for iOS)
- **Syntax typically doesn't matter**
- **Review helpful APIs**
  - String manipulation, popular data structures, searching and sorting algorithms, etc
- Be able to talk about why you picked that language

# Study CS fundamentals

- **String or array manipulation** - Great for tech screens
- **Linked lists** - Often used in whiteboard interviews because they expect you to draw pictures.
- **Trees** – BSTs, self balancing. Often used when building up directories or searching for something.
- **Hash Tables** - If you are organizing data for lookups... chances are the answer is a hash table.
- Graph traversal, BFS, DFS, Greedy, Dynamic programming, etc

Fortunately, there are several online materials to study them with coding



# Do's and don'ts

## **Do's**

- Explain what you are coding/typing to the interviewer, what you are trying to achieve
- Practice good coding style. Clear variable names, consistent operator spacing, proper indentation, etc
- Type/write at a reasonable speed
- As much as possible, write actual compilable code, not pseudocode
- Demonstrate mastery of your chosen programming language
- Etc

# Do's and don'ts

## **Don'ts**

- Remain quiet the whole time
- Spend too much time writing comments
- Use extremely verbose or single-character (unless they're common like `i`, `n`) variable names
- Copy and paste code without checking (e.g., variables need to be renamed)
- Write too big (takes up too much space) or too small (illegible) if on a whiteboard
- Etc

# Practice

- Treat the interview like a standardized test
- **Practice coding without an IDE/Compiler/Computer**
  - Working with a whiteboard is a great idea
- Practice coding and talking aloud at the same time

# Study materials

Listed items are small portions among what you can find on the internet

- [Cracking the Coding Interview](#)
  - The ultimate material that you have to look at first
- [LeetCode](#)
  - Lots of questions in real interviews
  - Online mock interviews, solutions, and statistics when you paid
- [GeeksforGeeks](#)
  - Good for studying not only interview questions but also other materials (e.g., details about languages)

# Study materials

- [Educative.io](https://www.educative.io)
  - Good for taking several well-structured courses (e.g., system design, algorithm, etc.)
- [Hackerrank](https://www.hackerrank.com)
  - Providing step-by-step prep
  - Used in several company interviews
- [BAEKJOON \(백준\)](https://www.baekjoon.co.kr)
  - Korean prep website not only for coding interviews but also for complex programming problems (e.g., ICPC)
- [Pramp](https://www.pramp.com)
  - Mock interview with arbitrary people around the world
  - We will use it for our prep sessions

# Coding interview example

## **Find the Kth Largest Element in an Array**

Given an integer array `nums` and an integer `k`, return *the kth largest element in the array*

Remember the previous slide! (slide num. 29)

- Steps 1 & 2. Repeat the Problem and Give Examples
- Step 3. Approach
- Step 4. Code
- Step 5. Test and Big O

# Steps 1 & 2. Repeat the problem and give examples

- Clarify questions also ask questions
  - *"OK, I'm just going to reread the question... So, given a list of numbers and a target number, ... Let's make an example with the array of integers like 3, 2 ... 4, and  $k = 2$ . Then, since... I would return an output of 5 in this case. Is that right?"*
- Example

Input: `nums = [3,2,1,5,6,4]`, `k = 2`  
Output: 5
- With example
  - You will naturally discuss data structures for the input and output
  - You can clarify whether or not you really understand the question
  - You will naturally come up with clarifying questions
    - *"Will  $k$  be less or equal to the array length?"*
    - *"Which value do I have to return when  $k$  is greater than the array length?"*
    - *"Will we have duplicated numbers?"*

# Steps 3. Approach

- Explain our solution without coding
- Conversation is extremely important
  - Silent is the worst thing
  - If you cannot find out a good approach at the beginning, start with the brute-force solution
- Drawing is good, especially for list, graph, tree problems
- If the interviewer gives you a crazy look or say something, you might want to rethink your solution
  - Interviewer will help you to find out the optimized answer
  - They do not provide a direct solution, but their comments usually have hints



## Steps 3. Approach

*"OK, so I think there's a couple different ways we can approach this one.*

*First, we could sort the entire array, ... This is kind of a brute force method, so it would take about  $O(N \log N)$  time.*

*Another thing we could do is create a max heap ... So that would take about  $O(N)$  time to create the heap, and then it takes  $O(\log N)$  time to pop the root of the heap ...  $O(N + K \log N)$  which is better than method 1.*

*...*

*How does that sound?"*

# Steps 4. Code

- Explain your code
  - Talk about your code when you write
- Don't forget boundary conditions
  - E.g., when the input array is empty
- Write TODO
  - E.g., "*# TODO: check for value k is less or equal to the length of list.*"
- Use helper functions if you have slightly repetitive code
  - Remember code review examples in previous sessions
  - You can leave those helper functions empty and fill out them later (discuss it with the interviewer)

# Steps 4. Code

- Example code using the standard sorting function (inefficient, but for the simplicity of our example)

```
def kthSmallest(nums, k):  
    # TODO: check nums length (it should be greater than 0)  
    # TODO: check k value (it should be greater than 0)  
    # TODO: check for value k is less or equal to the length of list.  
    # Sort the given list  
    nums.sort(reverse=True)  
  
    # Return k'th element in the sorted array  
    return nums[k - 1]
```

# Step 5. Test and big O

- Test
  - Test your solution step-by-step
  - Say the following thing to the interviewer!
    - *"If you're OK, I'd like to move on to test cases to make sure I didn't miss anything in the code."*
- Big O
  - You should find out the correct Big O notation for your answer
  - It should be same with the thing that you mentioned during the "Step 3. Approach" phase

# Step 5. Test and big O

Input: nums = [3,2,1,5,6,4], k = 2

Output: 5

```
nums = [3,2,1,5,6,4]
```

```
def kthSmallest(nums, k):
```

```
    # TODO: check nums length (it should be greater than 0)
```

```
    # TODO: check k value (it should be greater than 0)
```

```
    # TODO: check for value k is less or equal to the length of list.
```

```
    # Sort the given list
```

```
    nums.sort(reverse=True)
```

```
nums = [6,5,4,3,2,1]
```

```
    # Return k'th element in the sorted array
```

```
    return nums[k - 1]
```

```
nums[2 - 1] = 5
```

```
O(n log(n))  
when n = len(nums)
```

# Feedback

Consists of overall grade, raw notes, short feedbacks for each rubrics

Overall grade (example): rubric is between 1 ~ 6

- Rubric: 4
- Short description: "I accessed TC with L3 rubric. TC has solid understanding on algorithms and data structures, but TC's code has several minor flaws. And, TC did not check corner cases when doing verification."

# Feedback (raw notes)

## **Clarification**

After a short intro, starts on 03:00

TC asked about how to treat empty case

TC did not raise any questions about buffer overflow

## **Approach**

06:00

TC explained brute force approach

I gave hints with one simple 2x2 matrix

11:00

TC explained well-defined  $O(n^2)$  approach with examples

...

## **Code**

15:00

Start coding, made a signature

20:00

TC made code for case analysis, but code is a little bit redundant

...

## **Full transcripts**

...

# Feedback (rubrics)

Select one of following options for each rubric:

points 1 ~ 4

- **Comm. & Comprehension**  
3, TC shows great communication skills. ...
- **DS & Algorithm**  
3, TC shows great understanding on algorithm and time complexity
- **Coding**  
2, TC's code has errors and redundant parts. ...
- **Efficacy**  
2, when compared to other solid L3 candidates, TC couldn't not proceed follow-up questions, which is acceptable, but not solid.



# Conclusion

# Conclusion

We looked at

- Tech interview overview
  - What is tech interview?
  - Interview process and format
  - What do we need to do? – in general
- Coding interview
  - Coding interview process
  - What do we need to do? – specific for coding interview
  - Coding interview example - step by step
  - Coding interview example (videos)
  - Coding interview feedback example