App Academy

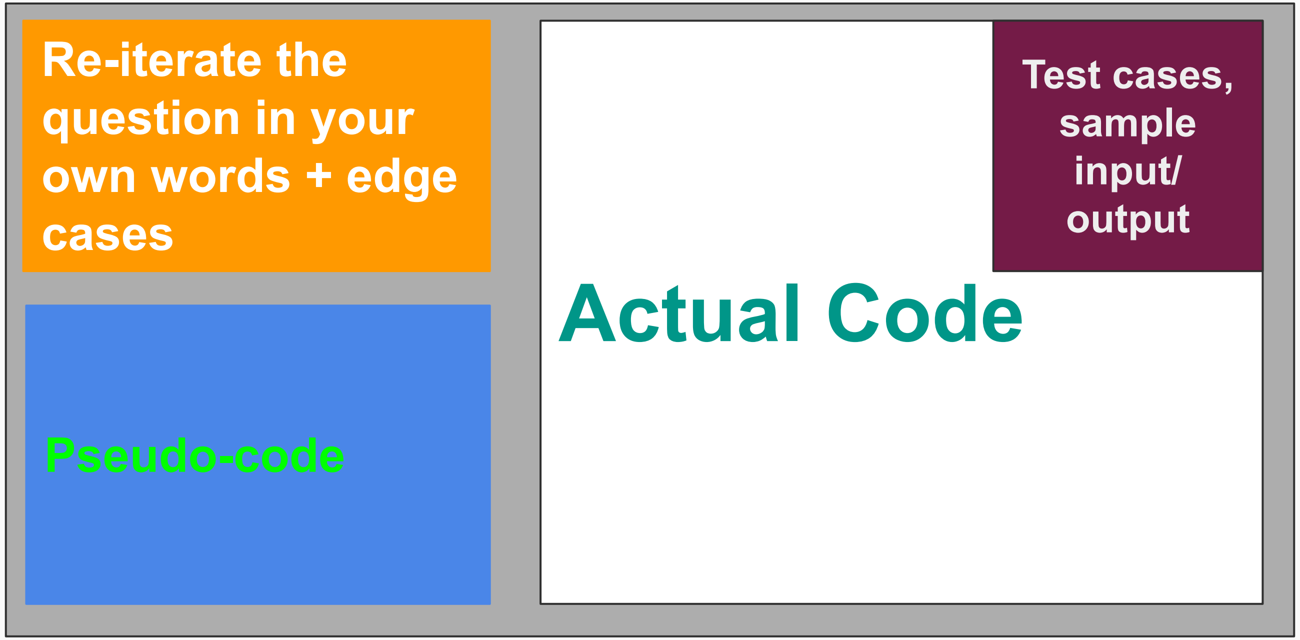
DS&A Interviewing Tactics

Succeeding in a DS&A interview

Six Steps (in four categories):

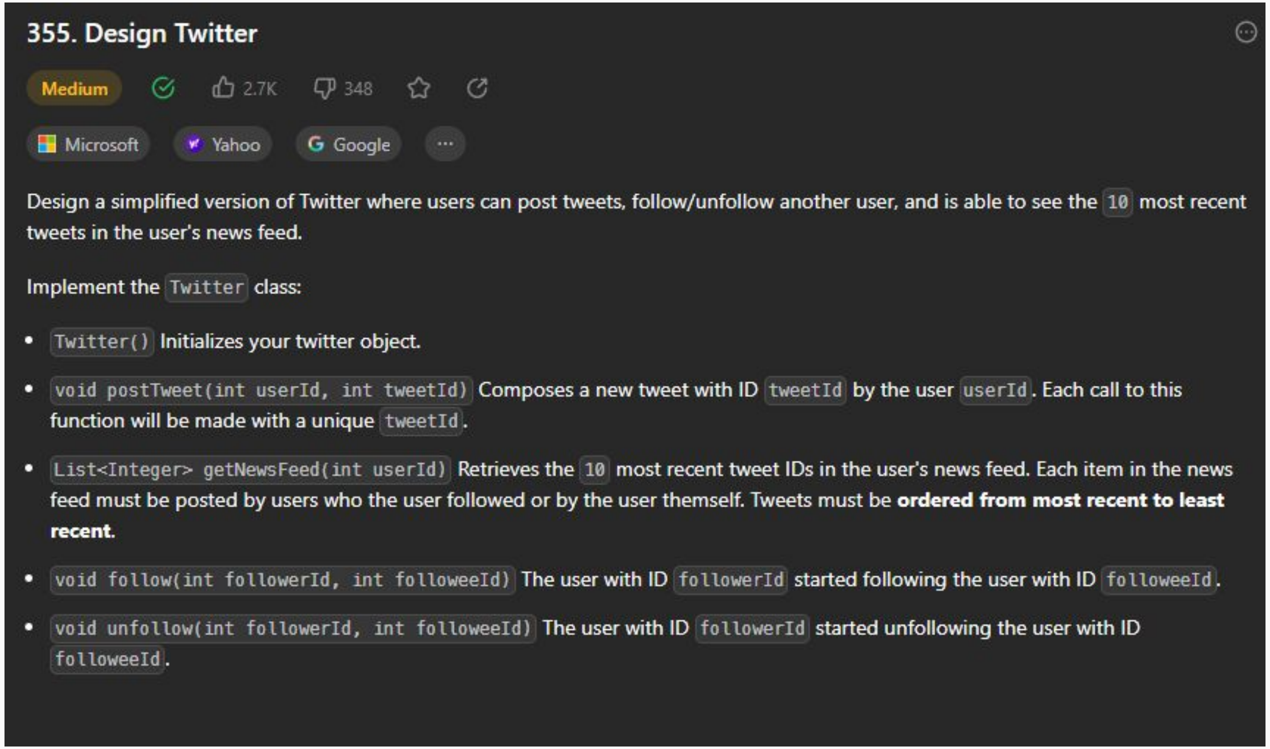
- Understand (30%)
  - 1. Ask clarifying questions (Do you understand the prompt?)
  - 2. Setup Sample I/O’s/Identify edge cases
- Plan (45%)
  - 3. Identify approach(es) (What DSs combine to solve problem? Complexities?)
  - 4. Pseudocode! (This is the most important step)
- Code (25%)
  - 5. Implement your solution (CODE)
- Test and Analyze (100%)
  - 6. Test your I/O’s and revisit Big-O Notation

In-Person Whiteboard Organization


Re-iterate the question in your own words + edge cases
Pseudo-code
Test cases, sample input/ output
Actual Code

You do not have to organize exactly like this, but make sure you stay organized in your own preferred way

Let's do a mock interview



Design a simplified version of Twitter where users can post tweets, follow/unfollow another user, and is able to see the `10` most recent tweets in the user's news feed.

Implement the `Twitter` class:

- `Twitter()` Initializes your twitter object.
- `void postTweet(int userId, int tweetId)` Composes a new tweet with ID `tweetId` by the user `userId`. Each call to this function will be made with a unique `tweetId`.
- `List<Integer> getNewsFeed(int userId)` Retrieves the `10` most recent tweet IDs in the user's news feed. Each item in the news feed must be posted by users who the user followed or by the user themself. Tweets must be **ordered from most recent to least recent**.
- `void follow(int followerId, int followeeId)` The user with ID `followerId` started following the user with ID `followeeId`.
- `void unfollow(int followerId, int followeeId)` The user with ID `followerId` started unfollowing the user with ID `followeeId`.

https://leetcode.com/problems/design-twitter/

Understand the Problem

Clarification questions

- Can we assume that all Tweets IDs will be unique?
- The example shows created Tweet IDs in order. Is this guaranteed?
- Where are each of our parameters defined?
  - userID, followerID, followeeID, tweetID
- How many calls can we expect to be made?
- Edge cases:
  - What would happen if a user tries to follow a userID that does not exist?
  - What if a user attempts to unfollow themselves?

Time to plan

What data structure would be useful here?

- Let's consider…
  - We need a way to store userIDs and tweetIDs
  - We need a way to create relationships between userIDs to show who is following who
  - We need a way to keep track of the time that tweets were made
  - We need to be able to easily access who a user is following based on their userID

How to get a list of followers based on userID?

- This sounds kinda like a key-value pair!
- We can create a hash map that contains all the users as keys, and their values could be a list of users that they follow
- How do we want to list out the followed user IDs? Array vs Set? Do we care about being able to use an index to find a specific followed user? Do we need duplicates?

```
this.users = {
        1: {3, 4}
        2: {1, 3,4}
        3: {1}
        4: {1, 2, 3}
}
```

How should we structure our tweets?

- What if we made a tweet an object?
  - Then we can easily keep track of of a tweetID and the associated userID
- How do we keep track of which tweets are posted before others? Are there any data structures that inherently allow us to track the order?
  - Arrays!

[{ user: 1, tweetId: 5 }, { user: 2, tweetId: 6 }, {user: 3, tweetId: 7}]

Let’s make sure we check all reqs

- postTweet
  - We can push tweets into a Tweets array
  - O(1)
- follow
  - We can add the followeeID into the follower’s set of followees
  - O(1)
- unfollow
  - We can delete the followeeID from the follower’s set of followees
  - O(1)
- getNewFeed
  - We can get filter the Tweets array by followed usedIDs, slice the last 10 tweets, and reverse it
  - O(3n) => O(n)

Let’s code it out

Code it out

https://leetcode.com/problems/design-twitter/description/

Walkthrough and Analyze

Walk through examples and revisit big O

- Edge cases:
  - What would happen if a user tries to follow a userID that does not exist?
  - What if a user attempts to unfollow themselves?
- Big O
  - Was our original analysis consistent with what we end up with?
- Alternative solutions
  - Are there any alternative ways to solve the problem that you can think of?
  - Trade-offs?

Questions?