

# What's next?

# Where have we come from?

- We've learned and practice a step by step strategy
- We've solved more than 30 of the most common practice problems
- We've built a consistent practice routine so that you can continue making progress
- We've built the confidence required to truly succeed

# Where do we go from here?

- Continue to practice and consistently improve
- Work to develop deeper skills and true mastery
- Start interviewing

# The biggest mistake you can make from here...

Don't wait to start interviewing

# Keep Studying and Develop Mastery



bytebyte

# What we've done so far

- This course focused on the most important 20%
- We focused on getting results as quickly as possible
- Now, you can start filling in the gaps

# General Strategy

- Implement the data structure and common algorithms
- Solve the most common problems using that data structure
- Test your knowledge and iterate
- Block practice + Randomized practice

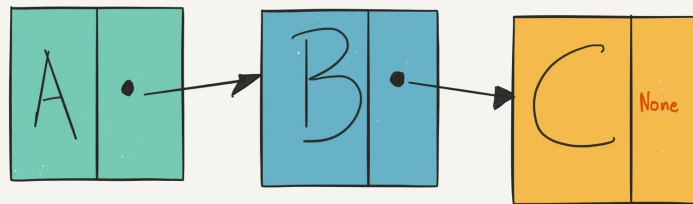
# Implement the data structure and common algorithms

- You should know how to implement the data structure and algorithms
  - Eg. Do you know how to remove an item from a heap?
  - Some basic operations are pretty tricky
- Basic algorithms
- Most problems can be solved with a combination of basic patterns



# Implement the data structure and common algorithms

- Eg. Linked Lists
  - Implement singly and doubly linked lists
  - Iterate over a list
  - Insert a node into a list
  - Remove a node from a list
  - Reverse a list
  - Find cycles in a list
  - Use runner pointers to find the nth element, split a list, etc.
- I've given you this in the Bonuses section



# Find the most common problems using that data structure

- Leetcode -> Sort by popularity
- Reddit thread on best “bang for your buck” problems
  - [https://www.reddit.com/r/cscareerquestions/comments/b4t12i/bang\\_for\\_your\\_buck\\_leetcode\\_study\\_guide\\_question/](https://www.reddit.com/r/cscareerquestions/comments/b4t12i/bang_for_your_buck_leetcode_study_guide_question/)

**Array**

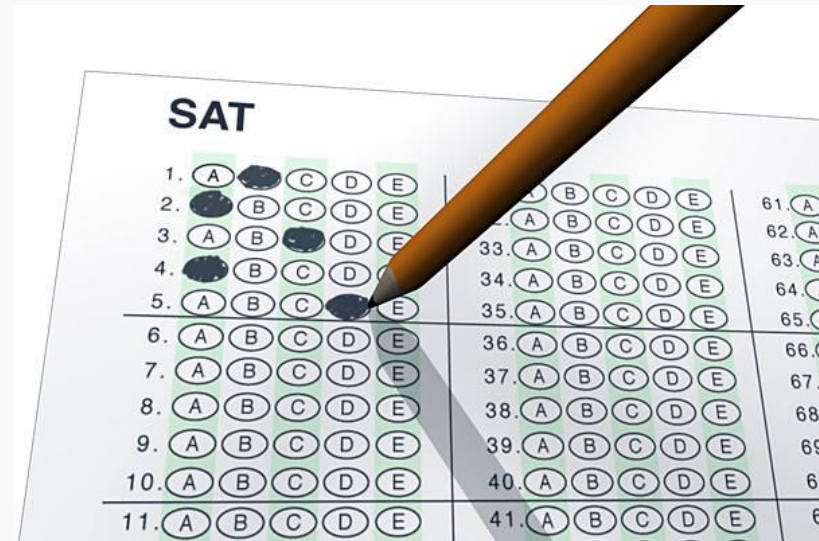
You have solved 1 / 167 problems.

Show problem tags

| #   | Title                              | Acceptance | Difficulty | Frequency |
|-----|------------------------------------|------------|------------|-----------|
| ✓ 1 | Two Sum                            | 43.6%      | Easy       |           |
| 4   | Median of Two Sorted Arrays        | 26.1%      | Hard       |           |
| 11  | Container With Most Water          | 44.0%      | Medium     |           |
| 53  | Maximum Subarray                   | 43.4%      | Easy       |           |
| 42  | Trapping Rain Water                | 42.8%      | Hard       |           |
| 561 | Array Partition I                  | 68.7%      | Easy       |           |
| 15  | 3Sum                               | 23.8%      | Medium     |           |
| 66  | Plus One                           | 41.1%      | Easy       |           |
| 121 | Best Time to Buy and Sell Stock    | 46.9%      | Easy       |           |
| 122 | Best Time to Buy and Sell Stock II | 51.6%      | Easy       |           |
| 169 | Majority Element                   | 52.3%      | Easy       |           |

# Test your knowledge and iterate

- Spaced repetition and recall are **CRITICAL** here
  - This is the part where you'll forget stuff if you're not careful
- Try to review challenging topics at least 3 days per week, even for 10 minutes
- If you get stuck, what is the pattern that you didn't understand?



# Block practice + Randomized practice

- What we're doing here is really block practice
  - Focusing on one skill
- Look for commonalities to help you identify these problems
- If you get stuck in randomized practice, are you unstuck when you know the DS?

```
int getRandomNumber()  
{  
    return 4; // chosen by fair dice roll.  
             // guaranteed to be random.  
}
```

# System Design

# What is System Design?

- Generally, designing software systems
- Basically “be a software engineer”
- More specifically? It depends

# What is System Design?

- Depending on the company/interviewer, you may be expected to:
  - Talk about systems at a high level
  - Develop APIs
  - Weigh pros and cons of different technology choices
  - Develop a data model
  - Develop a testing model
  - Or something else...

# So how the hell do you know what to do?

- Collaborate with your interviewer
- Work iteratively
- I'll share some strategies in a minute

# But first...

- Should you even be worrying about system design?
- If you're a new grad, no
  - Most places don't ask until you have 3-5 years experience
- Will the interview be make-or-break?
  - Doubtful
  - You already know how to do this stuff and there's no one right answer

# How to prepare?

Focus on:

1. Strengthening your existing knowledge
2. Having a clear gameplan

# Strengthening your existing knowledge

- You're already working as an engineer so you already know a lot of this stuff
- Focus on reviewing the tech that you're already familiar with. Make sure you understand the tradeoffs
- Only worry about learning brand new stuff if you are coming from a totally different background or filling in gaps

# Resources for strengthening knowledge

- [Hired In Tech System Design Course](#) (free)
- [Grokking the System Design Interview](#)  
(paid)

# Have a clear gameplan

- This is the same as what we've done for technical problems. You need to have a plan
- Generally:
  - a. Understand constraints/use case
  - b. Discuss at a high level
  - c. Ask them where to go deeper
  - d. Repeat

# Some miscellaneous tips

- Deeply understand the use case for what you're building
- Make sure to do what they ask but also try to highlight your skills
- You will be judged partially based on your past experience, so assume they will have reasonable expectations

# Behavioral Interviews

# What counts as a behavioral interview

- Anything that is not technical
- “Tell me about yourself”
- “Tell me about a time...”

# Biggest behavioral interview mistakes

- Not selling yourself
- Not preparing
- Rambling
- Not being genuine
- Not answering the question
- (Bonus) Saying stuff that makes you look bad

# How to prepare properly

1. Research the company
2. Fill out your interview prep grid ([Link](#))
3. Prepare a short description of each major project
4. Practice with the grid
5. Practice without the grid

# What if I don't have relevant experience?

- Get more creative
- We tend to focus very narrowly on the question being asked, but you have a life full of experiences

# How to structure your answers

- The STAR technique
- Situation (1-2 sentences)
- Task (1 sentence)
- Action (3-5 sentences)
- Result (3-5 sentences)

# More on behavioral interviews

- <https://www.byte-by-byte.com/behavioral-interviews/>