

WELCOME TO SE103!

Please make sure to **include your pod number in your zoom name (mentors included)**! E.g. “7 - Mari”

In Zoom: Go to Participants → More → Rename

Find out your pod number in the “**Groupings**” page in the course portal



Linked Lists



AGENDA & ANNOUNCEMENTS

- Linked Lists (30 mins)
- Breakout sessions w/ mentors (60 mins)
- Common Interview Practice Mistakes, feedback survey, questions (30 mins)
- *If you want me to see a question/comment **ask in the Slack help chat***
- *There will be Office Hours with Bashir on Monday 5pm PST*
- *Getting assignment help order of operations:*
 1. *Ask your pod teammates*
 2. *Ask your mentor*
 3. *Look at online resources (e.g. Leetcode discussion forums, Youtube, etc.)*
 4. *Post in the help chat with your problem and how you tried to solve it*

Linked Lists

LINKED LISTS

- Comprised of nodes that store an arbitrary value (usually a string or a number)
- Different types of linked lists
 - Singly-linked list - each node only has a *next* pointer
 - Doubly-linked list - each node has a *next* and *previous* pointer

Singly-linked List



Doubly-linked List



LINKED LIST CLASS

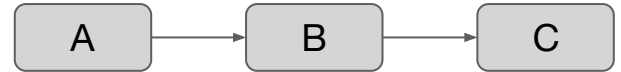
- You only need a value in order to initialize a node
 - *next* and *prev* (if doubly linked-list) are optional

```
1 ▼ class LinkedListNode:
2 ▼     def __init__(self, value):
3         self.next = None
4         self.value = value
```

CREATING LINKED LISTS

- Initialize each node and assign the next pointers properly

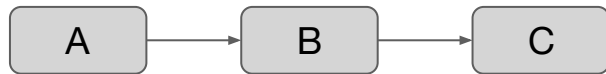
```
1 class LinkedListNode:
2     def __init__(self, value):
3         self.next = None
4         self.value = value
5
6 a = LinkedListNode('a')
7 b = LinkedListNode('b')
8 c = LinkedListNode('c')
9 a.next = b
10 b.next = c
```



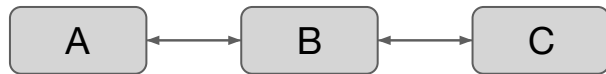
LINKED LISTS: THINGS TO KNOW

- All problems deal with pointer-manipulation and traversing the list
- Most problems require you to use $O(1)$ space
- Common patterns in solving these problems:
 - Dummy-head
 - Two-pointer
 - Multi-pass

Singly-linked List

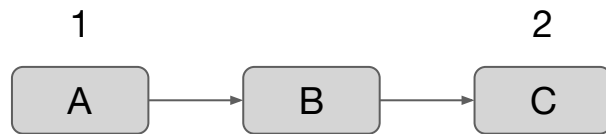


Doubly-linked List



TWO-POINTER

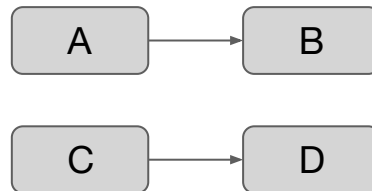
- Use two/multiple pointers to manipulate references to nodes (e.g. reversing a linked list)
- Also used to get/infer certain information about list (e.g. detect cycle)



DUMMY HEAD

- Create a 'dummy head' node and construct a list using its *next* pointer
- Return *dummyHead.next* to return the newly constructed list
- Very useful when you need to manipulate pointers and create a list with the same nodes but different ordering
- Also very useful for edge-cases

Interleave Two Lists



Dummy Head



OTHER PATTERNS

- Multi-pass
 - Useful if you can guarantee constant amount of passes or if you need to know the length of the list
- LL Reverse
 - Adding two lists together
 - LL Palindrome

REMOVE NTH NODE

- Remove the nth node from the end of the list
- [Leetcode Link](#)

Breakout Sessions

BREAKOUT SESSIONS

- Mentors, introduce yourselves to your pod!
- Swap Nodes in Pairs
- Rotate Linked List
- *If you finish early, do additional problems together*
- ***DM me if you don't have a mentor in your pod***

Common Interview Practice Mistakes

A LITTLE BACKGROUND ABOUT ME

- I was never naturally good at algorithmic interviews
- I failed a bunch of interviews and had impostor syndrome
- I lucked out on my intern interview and didn't get asked data structures/algorithms



THINGS TO KNOW ABOUT PROGRAMMING INTERVIEWS

- For the most part, it's a separate skill
- There are different types of interviews
- Data structures and algorithm interviews have a steep learning curve, but once you crack it then you will be able to ace a *good* amount of interviews



CATALIN PIT



@catalinmpit

Tech interview:

- Dynamic Programming
- Dijkstra Algorithm
- Linked Lists, Stacks, Queues, Binary Trees
- 10 hard LeetCode questions

Day-to-day job:

- Fix typo in README
- Center the form
- Fix the Twitter icon link
- Add secret_key

HOW YOU PRACTICE MATTERS

- How you practice has a huge impact on your performance
- Very few people practice effectively for interviews

*"Most people have the will to win,
few have the will to prepare to win."
- Bobby Knight*

COMMON MISTAKES WHEN PRACTICING FOR INTERVIEWS

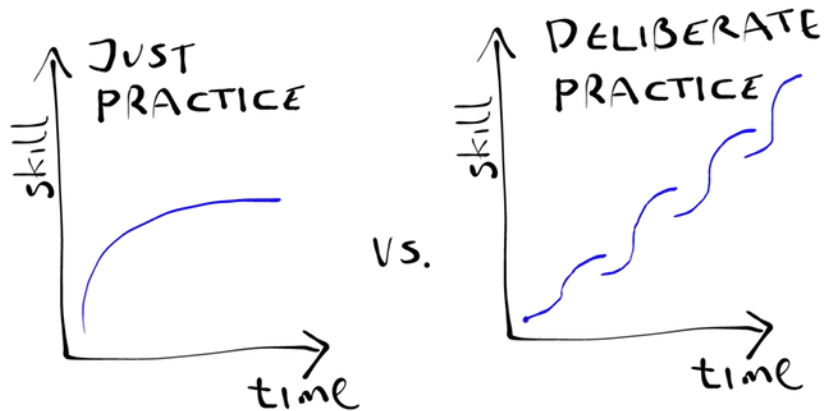
- What do you think are the most common mistakes people make when practicing for interviews?

COMMON MISTAKES

1. Mindlessly doing as many problems as you can
2. Forgetting about problems/patterns already seen
3. Not being emotionally prepared
4. Getting demotivated
5. Not preparing for other types of interviews

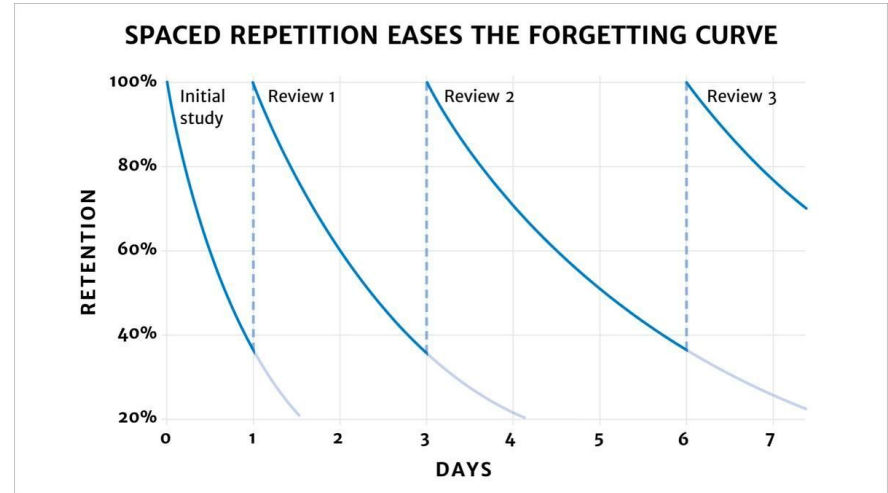
MINDLESSLY DOING AS MANY PROBLEMS AS YOU CAN

- Amount of Leetcode questions done != Amount of your understanding of the material
- Brute-Memorizing solutions is not efficient
- Add **deliberate practice** to your regimen
 - "...a special type of practice that is purposeful and systematic"



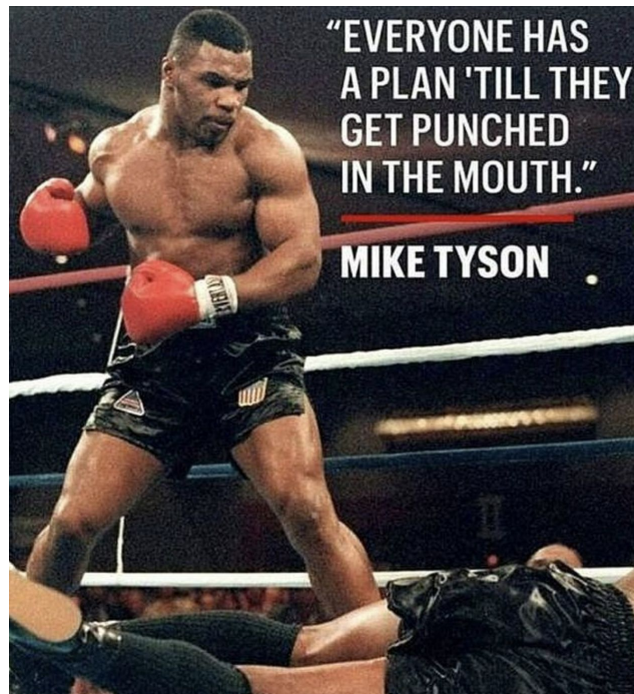
FORGETTING ABOUT PROBLEMS/PATTERNS ALREADY SEEN

- Acing interviews is largely on recognizing a pattern/solution you've previously used and applying it to the problem at hand
- Fast recall is important during interviews
- Add ***spaced repetition*** to your regimen to retain as much information as possible
 - Use *Anki* to create custom flashcards with built-in spaced repetition



NOT BEING EMOTIONALLY PREPARED

- The emotional and physical (pre-COVID) environment during interviews are different
- Simulate conditions similar to an actual interview:
 - time limit, no compiler, no autocomplete
 - practice interviews with others (e.g. Pramp)



GETTING DEMOTIVATED

- Doing all hard problems is a recipe for demotivation, and ultimately failure
- Interleave easy-medium-hard problems together to gain confidence and challenge yourself
- Stuck in a problem? Look at the solution, then redo it later

*"A competitor needs to be process-oriented, always looking for stronger opponents to spur growth, **but it is also important to keep on winning enough to maintain confidence**" - The Art of Learning: A Journey in the Pursuit of Excellence*

TODAY'S RECAP

- Using UMPIRE in problems
- Common linked list patterns
- How to practice effectively

See you next time!

Please take < 3 mins to complete our [feedback survey here](#).

This helps us improve the class for you!