process.

My advice to people in the same position as I was in

- Don't be demotivated at your current level. I started from a place where I was bombing every interview and now I've done much better than what I was expecting. Start and be consistent.
- Try getting comfortable in every topic. Don't leave anything behind.
- Don't spend more than an hour on any question. If you can't figure out the solution, mark it and revisit later.
- Use the discuss forums. They are literally the best part about Leetcode and a key differentiator why no other platform can match up to Leetcode in terms of interview prep.
- Upsolve when you can't solve a problem - i.e look at the solution, understand and then do it again on your own.
- Keep taking notes about what a problem is teaching you. Keep revisiting them. Spaced repitition helps in committing things to memory.
- Don't be in a hurry. Enjoy the ride :)

Cheers! (also thanks to Leetcode and the amazing guys who keep posting epic stuff in the forums! I owe a lot to all of you)

Tips and Resources for System Design

I'm updating the post with the resources I used for system design interviews as some of the fellow leetcoders wanted to know.
First of all, depending on your experience, the expectation in system design round varies. If you are interviewing for SDE1/entry level SDE role, most companies don't bother asking system design. As the level progresses, you will be evaluated on things like the breadth and depth you possess, the variety of approaches you are able to take and trade-offs you are able to make etc.

- On to the preparation part - clearly, experience is the best teacher in this area. But for beginners, I would suggest going through System Design Primer. This has been the single best resource for me to lay down my basics. If you have time, I would highly recommend going through every link listed under references in the repo. Highly useful!

- Second, I purchased a course on Systems Design Interview. The theory part of this is not great, but the questions cover a lot of variety and take a very structured approach. This helped me a lot! Honestly the most return on investment for system design. If you are done with these two steps, you should be comfortable enough to crack most system design interviews.

- Third, as an added bonus, you can go through some of the seminal research papers in the field of distributed systems. These discuss the most fundamental ideas and trade-offs in designing large scale systems. It is not necessary to do this to pass the system design round, but it will clearly spike your curiosity and lay down a much stronger foundation:

    - Amazon - Dynamo paper
    - Google - Map-reduce paper
    - Google - GFS paper
    - Facebook - TAO paper

Everything from here on is for people targeting senior engineering positions

The next steps, for people who are applying for more senior positions (8-10+ years of experience, for example) would be to demonstrate a larger depth and breadth of the area. The best way to gain this is by deepening your theoretical understanding and practical understanding.

- For deepening theoretical understanding, I found the following two resources immensely helpful:

    1. Distributed Systems for Fun and Profit - free book, 100 pages or so, but excellent content
    2. Designing Data Intensive Applications by Martin Kleppman - this is a more in-depth book, and you can get it on Amazon/libgen

- For deepening practical understanding, going through companies' engineering blogs is the best resource.

    - Netflix tech blog, Uber, Twitter, Airbnb engineering blogs are high quality and are constantly updated.
    - For a consolidated list of company-wise engineering blogs, refer to this awesome github repository
    - There are a lot of great Youtube videos (append to youtube.com).
        - Jeff Dean's talk at Stanford: `/watch?v=modXC5IWTJI`
        - Building Billion user Load Balancer at Facebook: `/watch?v=bxhYNfFeVF4`
        - Netflix Guide to Microservices: `/watch?v=CZ3wIuvmHeM`
        - Amazon DynamoDB deep dive: `/watch?v=HaEPXoXVf2k`

Hope this helps!