

Amazon Interview Experience for SDE-1

- Difficulty Level : [Medium](#)
- Last Updated : 20 Aug, 2021

Got an offer from Amazon! Here is my experience.

I do not have a degree in CS and learned on my own (no Bootcamp). You can achieve anything you want to if you work really hard. I had several portfolio projects on my resume and was contacted by a recruiter.

Technical screen (online):

- Find the two smallest numbers in an array
<https://practice.geeksforgeeks.org/problems/find-the-smallest-and-second-smallest-element-in-an-array3226/1>
- Given a max distance, find the two numbers that added together are closest to the max distance

I must have done well because I skipped the phone screen and went on to the final interview round. I had an option of scheduling two weeks later and 5 weeks later. I didn't know data structures/algorithms prior to this and chose the slot 5 weeks later so I could have time to learn them.

Final interview questions:

- <https://practice.geeksforgeeks.org/problems/longest-common-prefix-in-an-array5129/1>
- Maze traversal
- System design: What kind of classes and object-oriented design would you create to take in CSV, tsv, and JSON objects, parse them, and enter them into a database? (This was a very hard question for me, but I hadn't researched system design a ton because I knew it wasn't a deal-breaker for an SDE-1).
- Behavioral: they will grill you about your projects. Research the LPs and come up with a list of ~10-12 situations that fit them. Rehearse them. Look up common amazon LP questions and make sure you can answer them. They will grill you about your projects too. What would you have done differently? Be prepared to dive deep into them.

I did well on most questions (I was able to answer them) and struggled on the last one.

Advice (especially helpful for self-taught programmers who don't know DS/Algo well):

- I studied ~9 hours every single day. Make sure you know all the data structures on the cheat sheet they send you. Start watching youtube videos, and when you see related topics, write those down. Have a running list of all the topics you need to know. Have a running list of all of your questions. Make sure you go through them ALL.
- Do several Leetcode problems a day. Spend 45 min trying to solve it before you look up the solution. Make sure you actually understand the solution even if it takes a few days. Ask your friends to help you. Come back to those questions a week later and see if you can solve them.
- Figure out the biggest priority topics by looking at recent interview experiences on GeekforGeeks/elsewhere. I saw that trie questions had been recently asked and knew this was a priority. I also knew it was hard for me to solve maze problems, and that they do get asked, so that was a priority too.
- Take notes on EVERYTHING. Go through your notes and write a review. What are a few bullet

points of how you solve each problem? That way when it comes time to interview, you can remember \xe2\x80\x9cI create a recursive function for this that takes in __\xe2\x80\x9d.

- Do NOT get discouraged. I thought for sure I wouldn\x9t get it because I couldn\x9t get any of the problems on the first try, and though they were very hard to learn and understand. The difference is, in the interview, they give you hints along the way. So if you get stuck, you\x9ll be able to keep moving forward. You can do this!!!
- Don\x9t be defensive in the interview. Check your ego. Let them give you suggestions and *take* them. It\x9s a dialogue with your interviewer.
- Try to learn as much as possible while being able to retain that knowledge. Take care of yourself, but push yourself up to that limit, and know that your interview experience can dictate your salary too.
- You. Can. Do. This. It will be so frustrating at times along the path but that\x9s no indicator of the final result. Good luck!!!

My Personal Notes\narrow_drop_up

Add your personal notes here

Save