1-interview-introduction

Contents

Introduction

To crack the coding interview you need to prepare with real interview questions. You must practice on real problems and learn their patterns. It is about developing a fresh algorithm, not memorizing existing problems

The interview

The interview is short, you might not be able to get through more than a couple of questions / tasks in the span of an hour or 45 mins. Do your best to explain your process while you try to solve the problem at hand. Talk out loud and explain your though process. The whiteboard part of the interview should not be pseudo code but rather a very shrunk down version of the real thing. Make as few errors as possible while writing the solution in code.

1. Analytical - did you need much help solving hte problem. How optimal was your solution, how long did it take you to arrive at a solution, if you had to design a new solution, did you structure the problem

well and thing through the trade offs.

2. Coding - were you able to successfully translate your algorithm to reasonable code ? Was it clean and well organized ? Did you think about potential errors, good style
3. Technical knowledge - do you have a string foundation in computer science and the relevant technologies
4. Experience - have you made good technical decisions in the past, have you built cool projects in the past, have you shown drive and initiative
5. Culture fit - do you have good communication skills and do your values fit with the company and team

The Questions

The questions are not pre selected, each interviewer selects their own questions, possibly even on the spot. Some companies like solution based questions, which involve algorithms, other focus on knowledge based questions, about technologies. The way questions are evaluated and their answers is based on all candidates they have been asked to. So it is relative, your performance depends on how well or unwell other candidates answered the same questions you have been asked. Experienced or novice candidates will often get the same type of problem solving questions, some companies might hold the experienced candidates to a lower standard, but that is not always given. If you have experience with enterprise systems and frameworks, but are not applying for such a role, you might have to show more technical skill to compensate for that. When answering questions, focus on yourself, not the team, tell them what you have accomplished, not the team.

When talking about yourself, try to answer the following questions

```
- What is your current role in the company you are/were at - be brief
- College / university background - what did you study, how did it go
- Post education - brief history of your career, taken from the resume
- Outside of work - what do you do, do you do projects in your spare time
- Hobbies (something unique) - best if the hobby is technical
```

The process

1. Screening interview - over the phone or chat, might involve some problem solving or whiteboard problems. Do not take it lightly. Some companies might have online shared editors, or documents, some might want you to write the problem on paper and read it back to them. The screening interviews might be from 1 to 3.
2. On site - in person interviews, from 3 to 6, some might be non technical, like over lunch etc. After that in the span of a week or two the candidate should expect some sort of feedback, more so if he is hired. No communication does not mean you have been rejected.

The resume

1. Keep it short, add the most impressive roles in your career thus far, the ones that will make you stand out. Add the most relevant technologies used by you and your team.
2. For each role, try to list your accomplishments, using the format - accomplished x by implementing y which led to z
3. Avoid listing too many technologies, or languages, you might not feel comfortable with, shrink the list down to the ones you like the most and feel the best in.
4. Try to avoid any grammatical mistakes in the resume, some employers might throw away a resume just because it has them.

5. Be careful with listing certifications, some companies might not like if a candidate has a certificate for specific technology or language, especially if the position you apply for is not relevant to that.

The projects

Include at least 2 to 4 projects, which you deem are your most significant accomplishments, State what the project was and which languages or technologies it employed. Do not add too many projects, you have ever done. Have a deep understanding of the technology and stack of the projects you have developed and displayed / demonstrated to them, in the resume. Make sure they can't catch you off guard with questions in regards to your own projects. Try to explain your projects, but do not go into too much detail, stick to the major points.

The interviewer

There are some questions to ask them as well. Here are some, which might give you some better understanding of what it is like to work at the company

1. What is the ration of testers to developers. What is the interaction like, how does project planning goes ?
2. What brought you to this company ? What has been most challenging for you ?
3. I noticed that you use technology X, how do you handle problem Y ?
4. Why did the product choose to use the X technology instead of the Y one, I know it has benefits like ABC, but many companies choose not to use it because of D.
5. Ask them what opportunities are there in the company to learn about a technology or a stack you are passionate about.
6. Ask about technology you are not familiar with, ask them how they use it, in their projects