kzr.buet08@gmail.com, zubaer.kh@gmail.com
Last update: 2018-01-08
https://goo.gl/rZQUVX

Stages of Tech Interviews

Resume

- Be truthful about the information in here, focus on your strengths.
- No need to make it long, one page is good enough in most cases. More than 2 pages is overkill.
- If you are a fresh grad, list major projects done as part of coursework. i.e. “Inventory management system for departmental store: Designed and developed an online inventory management system which provides predictions of future demand from past statistics. Used: MySQL, Java, Javascript.”
- Be prepared to face questions that is listed in resume. Ask yourself questions about your resume, and make some high-level answers beforehand. You need to give high level overview first, and then if the interviewer is interested in more details, s/he will ask follow-up question. i.e. if you say that you have designed “Inventory management system” as described on the previous bullet, you may be asked:
  - How did you predicted future demand from past statistics?
  - Why did you choose MySQL?
- List your non-academic achievements as well. A few examples, but not limited to:
  - Contributing to an open source project.
  - Participating in programming contests.
  - Participating in debate, game etc...

Phone Interview

- The interviewer will call you over phone, and also share a Google doc. So make sure you have an internet connection, and have a phone with good network, so that you can hear the voice clear.
- The phone interview should be easy. Do not take pressure. The interviewer will at first ask you a few warm-up questions preferably from your resume.
- After that he would probably give you a problem and probably ask you to code it (or part of it). A few bullet points to remember in here:
  - Take 1-2 more minutes to understand the problem correctly, and ask as many questions you need to clarify the question (they will judge you on this, whether you are understanding the problem correctly or just jumping into solution/coding). When I interviewed, I usually wrote down the problem in shared Google doc, and asked the interviewer whether I have understood the problem correctly. And also clarified with some examples.
  - Feel free to take some time to think about how you can solve it. It is better if you give the interviewer a heads-up about this, as when speaking over phone, it’s sometimes hard to understand whether you are silent as you are thinking, or the

line has cut-off: “Please give me a few minutes to think about how I can solve the problem.”

    - It's okay to take time to start coding, there is no need that you always have to continue talking (silence is totally alright). More importantly, if you have hard time in understanding the interviewer, you can tell him to write down the question. I personally took notes on the shared doc (they will share a google doc for the phone interview, where you will code), on the problem he is asking, and verified with him what I have understood before coding. You can also ask for an example, if he does not give you one, or you can make an example yourself and ask him whether the input and output you suggested are correct.

    - the best approach is thinking out loud. it is better to produce ONE solution as quick as possible (which may not be the best solution) than trying to figure out the BEST solution silently.
      talk about the time and space complexity of the current solution and then go to other solution where you will improve either time or space or both.
      Finally, always test your code.

Onsite Interview

Same as phone interview, with somewhat more involved problems.

[collected]

My Personal Experience [Kaysar]

I faced interview in two rounds.

First Round

- Soon after BSc.
- I have been participating in ACM ICPC style programming contests.
- Qualified for the onsite interview of Google, FB, Amazon, and Booking.com just facing some algorithmic questions in phone interviews.
- I took almost zero interview aligned preparation for onsite, especially for system design interviews.
- I didn't give any mock interview, I had trouble to make interviewer understand what I was thinking.
- I failed miserably at all of the onsites.

Second Round

- After my MSc.
- I lost 70% of my ACM ICPC style problem-solving skills.
- Qualified for the onsite interview of Google, and Booking.com solving mid difficulty level algorithmic problems: which are no harder than the online problems we face in our algorithm labs. I will be sharing the resources that helped me to pass this stage.
- I took interview aligned preparation which is different from ACM ICPC.
- I gave priority to system design preparation which I struggled last time.
- I gave at least 10-15 mock interviews. I asked seniors who are working at tech giants to take system interviews and give me feedback. I worked on their feedback.

Lesson

- ACM ICPC style programming contest background is not a must. In the industry, I would say at most 10% are from that background. What you need is confidence and problem-solving skill. you just need proper preparation plan and resources.
- If you get a call make it count. Give your everything.
- Give some focus on System Design Preparation.
- Mock, Mock, Mock interview is the key!
- They don't look for a scientist who can solve everything, they look for a team player. So, whatever you do make sure they are getting your point. Give them the impression that, you have the attitude.

Sample Question

will add soon!

Preparation Plan

Phone Interview

- Finish reading **Cracking the Coding Interview**: this will brush up your algorithm and data structure knowledge.
- Solve categorized problems from https://www.interviewbit.com/. From each category solve 2 to 3 problems. You will finish them before on-site.
- Solve the 150 easy and medium difficulty problems from https://leetcode.com. Make sure you pass all the corner cases.
- Remember difficulty of the problems will be such that you will be able to think about the solution within 10 min and code it quickly. So, focus on easy and medium problems for this stage.
- appear at least 3-4 mock interviews at https://www.pramp.com
- appear codeforces online programming contest, it will increase your speed.

Onsite

Coding

- Finish https://www.interviewbit.com/, this site has a small collection of good quality problems.
- Do as much as possible medium and hard problem from https://leetcode.com. If you aren't going to target, google you can skip practicing the hard problems. Buy a premium account just before one month of your onsite. You will get a lot of resources. Remeber, practice matters. Whatever you face in the interview, there's a high chance that you will be able to relate it to the problems you are practicing.
- Do write code on paper. Module your code using function and use reasonable naming. There's no value of your code if it's not readable to others.
- Practice taking while you are coding.
- Do the testing. Handle the corner cases.
- appear at least 10 mock interviews at https://www.pramp.com
- appear codeforces online programming contest, it will increase your speed

System Design

Step 1:

- This site is a gem: https://www.hiredintech.com/courses/system-design. Cover it a to z.
- http://cs75.tv/2012/summer/#about **only** Lecture 0 and 9.

Step 2:

- https://www.interviewbit.com/courses/system-design/topics/storage-scalability/
- http://blog.gainlo.co/?utm_source=Gainlo&utm_medium=header&utm_campaign=Gainlo
- http://blog.gainlo.co/index.php/2015/12/22/how-to-crack-the-coding-interview-with-practicing-only-30-questions/
- http://blog.gainlo.co/index.php/category/system-design-interview-questions

Step 3:

- Take the popular product like Gmail, gdoc, youtube, google search, web crawler etc. Design them by yourself.
- Manage people who are working at tech giant to take 3 system design mock interviews.

Important Links

- https://sites.google.com/site/steveyegge2/five-essential-phone-screen-questions
- https://docs.google.com/document/d/1fl9FRa_M2SD7yNIn2v04nW0Fd2TqIlsuShTyG9qxEtE
- https://docs.google.com/document/d/1bYGvFQuBp4i5hOUjHbKA2mySXahBea9VxdyI_NEDeNY
- https://highlyscalable.wordpress.com/2012/02/01/mapreduce-patterns/
- http://infolab.stanford.edu/~backrub/google.htm
- Paid sys design materials https://www.educative.io/collection/5668639101419520/5649050225344512