



A to Z- guide to your campus placements

Sep 23, 2020



Welcome to the A to Z- guide for cracking your CDC Placements, (or any coding interview preparation for that matter). This post has been due a long time, and I am glad, I am finally publishing this. Please note that this guide is not exclusive, and is only based on my personal experience, and those of my friends, who appeared in the placement process in December 2019. I have tried to cover almost everything that I experienced in the journey, but might obviously have missed some important things, and hence I don't claim this guide to be A-Z but instead A-Z- (approaching Z from the left).

My Background

I majored in Mathematics and Computing from Indian Institute of Technology, Kharagpur in 2020. I had a chance to appear in the placement session of 2019, and got successfully placed in Cohesity. Currently, I am working at Cohesity in the field of Distributed File Systems.

Who is this blog for?

I believe that this blog will be helpful to everyone preparing for a profile in Software Development, but people preparing for Machine Learning/Data Science or Quant roles can give this a read too. I have listed all relevant resources at the end, for each of the 3 profiles separately.

Also, it is important to mention that prior to 5 months before my interview, I had no knowledge of Data Structures/Algorithms whatsoever, other than the basic PDS course that I did in my 1st year. So, I will write this blog keeping in mind that you haven't had prior exposure to Algo-DS too, and how to prepare for this from scratch.

Crux

All right, without further ado, let's directly cut to the chase. I have divided my blog (and the whole preparation timeline) to 3 phases. Phase I extends from the time you start your preparation to 1.5 months (or, 1 month if you don't have enough time) prior to your interview. Phase II lies from 1.5 months (or, 1 month) before your interview to 2 days prior to your interview. Phase III is the grand finale, the 2 days prior to your interview to the time you had an offer in your hand.

Phase I

A good time to start your placement preparation is 5 months prior to your interview, and so, here I will consider the start of Phase I to be this date. If you happen to start the preparation late, do not worry. You can skip a few things that are mentioned here, and plan it accordingly.

Coding

The three most important resources for anyone to prepare for a coding profile are Geeks for Geeks, LeetCode and InterviewBit. I will suggest you to use GeeksForGeeks as a handbook and your go-to guide for understanding the theory and looking at standard solutions. LeetCode and InterviewBit should be your go-to point for solving questions.

1. Geeks For Geeks

Go through GeeksForGeeks thoroughly, especially Data Structures and Algorithms. Read about linked-lists, stacks and queues, and trees. Each of these topics have a ton of questions listed. Go through some of the questions and their solutions for each topic. You will understand how problems are solved. Do not attempt to solve them on your own, just go through to get an overview of the data structure or the algorithm.

Keep visiting this website whenever you are stuck, or your concept regarding a certain data-structure/algorithm isn't clear.

2. Interview Bit

InterviewBit is the best place to start solving questions. All questions are divided into certain sections, and it helps you to do an efficient topic-wise preparation. It starts with easier topics like Arrays and Linked Lists and then proceeds to Trees and DP and ends

with Graphs. Each section has a short tutorial to help you clear the syntax and other basic concepts. InterviewBit questions are more focused to each topic, and this will help you cover each topic one by one. This is especially important when you are starting out, and you may not know which questions belong to which topic. So, I will strongly recommend to finish InterviewBit at least once. Aim to have a score > 25k.

Tip: Don't get discouraged while you are solving questions from 'Arrays'. The questions in that section are tougher than usual. You will find it easy once you move down to Linked Lists and Trees.

3. LeetCode

LeetCode has a lot more questions (~ 1.5k) compared to InterviewBit (~ 0.4k). You obviously don't need to solve all of them. I suggest, to try to solve at least 150-200 questions (50 easy, 100 medium, 30 hard). You will find most of the questions already solved in InterviewBit, and this will be a revision for you.

LeetCode has a different performance measuring system than InterviewBit. In LeetCode, you should strive for having greater than 80 percentile score on your time and space.

LeetCode is the best place to find the best solutions to any question. Just head over to 'Discussion' section of a question, and find the solutions that are voted best. Going through these will help you improve the quality of your code, and the way you approach a problem.

- Top 200 LeetCode questions: The problem with LeetCode questions is you don't know which questions to solve from this collection! For this, I have prepared a list of all LeetCode questions arranged in order of likes. More liked a question is, more popular and more frequently it is asked in an interview. Aim to solve the questions in this order, and reach as far as you can.. (Maybe 200?)
- LeetCode weekly contest: Once you have a good hold on concepts, and have solved a good number of questions ranging from all topics, you should start to have a feel of what actual tests are like. LeetCode weekly contest is a good way to solve new questions and with the timer running.

Some points to note:

- Dynamic Programming is the most important topic to prepare among all. Pay special attention to this, and solve as many questions as you can related to DP.
- Questions from Binary Search can sometimes be tricky, and be applied in unusual places. You will/have come across Painter's Partition Problem.
- Topological Sort and Union-find are quite important in Graphs.
- Trie is another important topic to look into. This may be a bit advance, but is important for interviews.
- Binary Index Trees and Segment Trees are quite advance, and will not generally be asked in an interview. (Except, maybe Google?)

Machine Coding

Some companies in their interviews ask you to design simple applications on a whiteboard. You are expected to know how Object Oriented system works, and how to write modular code. For practice, you can try designing Splitwise, Trello or Snake and Ladder.

Have a look at this [website](#) to get some idea about what is expected of you in a machine coding round.

Machine Learning/ Data Science

For preparing for Data Science roles, there is no one fixed source. My recommendation is to prepare a list of topics to cover, and study each topic separately. The amount of depth you should go into for each topic varies based on the company. Some companies expect you to understand the derivations behind each equation governing the certain technique and some companies will only care whether you know the technique and how to apply them in practice.

A good rule of hand, is to go in as much depth as you can, *in those topics which you have mentioned in your resume*. For example, if you have mentioned SVM in your resume, you are expected to know what margins and kernel trick are (with, the maths governing it!). For things you have not mentioned in your resume, you should have a breadth of knowledge, and should know which techniques are to be applied in which cases, and the various drawbacks and advantages of the techniques. To achieve a *breadth of knowledge*, my recommendation is to read as many ML-related blogs as possible.

Another important point to note is that companies coming for 'Data Scientist' roles, do **not** necessarily ask questions from Data Science in their shortlisting test. Almost all companies coming for Data Science profiles (except Microsoft DS, in our year) had a coding round too where questions were asked from Algo/DS. Hence it is important for you to get your coding strong, as this is almost always the shortlisting criteria. Data Science knowledge is more of a requirement during your interview.

Phase II

You have approximately 1.5 months to go before you face your interviewer. Consider this phase to start, when your placement tests are about to start, or have just started. This is the time to focus on how to crack your placement tests primarily. Once all of your placement tests are over, you should start preparing for interviews.

1. Placement tests

Once tests begin, stop solving new questions from LeetCode. This is the time to:

- Revise all standard questions/concepts. Revise the notes that you prepared from your Phase I. Go through questions that you might have bookmarked, or you wanted to take another look.
- Solve questions from Placement Docs. You will be surprised to see the previous year questions that company asked. You will have seen none of them in LeetCode/InterviewBit, and this may seem a bit scary. But don't worry. Questions that are asked in placement tests are generally new, which you haven't seen before. Going through previous year questions will help you have an idea on what kind of questions to expect in test. Placement Docs are regularly updated in this facebook group.

2. Interview preparation

Once all tests are over, its time to start preparing for your interview.

Prepare your Resume

- I reiterate *Prepare your resume very very well*. This is one part, that can *break* your interview. Not all companies have a specific 'Resume round', but your interviewer might glance at your resume in a coding round and throw you a question based on one of your projects that you did, or a course that you took. Remember, you can afford to not answer a coding question but you *cannot* afford to not answer a question based on your internship/project. That puts a bad impression on the interviewer.
- Prepare short notes based on each project/internship that you did. Go back to your notes from the internship or the presentation, and study them well. Understand all techniques that you used at that time. Prepare a *critical analysis* on why you chose to go with a certain technique and not with something else. Revise all courses that you mentioned on your resume.

Stop coding; Start solving questions on paper

- Start practising solving questions on paper. You will notice that this is not entirely same as writing code on a code editor. You will have to think of the solution beforehand, have a clear understanding of the approach, and only then start converting it on paper. Try to keep the code as clean as possible: making it messy implies you didn't have a clear approach in your mind when you started. Also, *no psuedo code*. Write the code in proper syntax.

InterviewBit Mock Interview

- InterviewBit also has a Mock Interview feature where you can schedule interviews with a stranger, and you take turns interviewing each other. This proves to be fruitful in getting an actual feel of the interview.

Basic Data-Structures ramp-up

- Go through the Data Structures from Geeks for Geeks again. Do you remember how to implement QuickSort or a Priority-Queue? The implementations of basic Data-structures or algorithms are rarely used while coding, since you directly use STL

library. But, it is important to brush them up, since these can easily be asked in an interview.

Geeks For Geeks Interview Questions

- Go through common company-wise interview questions from Geeks for Geeks. During the hiring season, common ongoing interview questions are also updated in LeetCode's Discussion section.

Phase III

Welcome to the grand finale! Now is the time to stop learning new things, and revise whatever you have learnt up till this point.

Before Interview

Prepare a short note on basic interview questions like your introduction, your strength and weaknesses etc. These are generic questions, and are not usually asked in Software development profile, but it is always good to be prepared.

Brush up your resume well. Form a peer group, and take turns asking each other questions from the resume. Try explaining each of your projects from scratch to a friend, and see if you are able to do a good job.

Revise your notes, and go through concepts/formulae/important questions that you have noted up till this point.

During Interview

The interview days can be quite tiring. Be prepared. Have a good night's sleep because the next days can be very tiring. Plan your companies beforehand according to your shortlists. If you have more than 2 shortlists in a company in one half of the day, you will have to prioritize. Talk to placecoms about your preference, and they will try their best to accomodate you accordingly.

Also, once you have an offer from a company, you *cannot appear for an interview for any other company*, even if it is in the same day or the same half. Therefore, prioritizing becomes quite important. You do not want to sit for an interview, just "for trying it out". If you happen to get selected, you cannot back down. (*Specific to KGP*)

The most important factor to keep in mind at this point is, *Hope for the best, prepare for the worst*. Give your 100% in the interview, but do not get disappointed if you happen to get rejected. There will be more interviews to come, hence it is important to maintain your composure. Do not get disheartened.

Some Tips and Tricks

- When placement tests are ongoing, keep in touch with your friends from other IITs/NITs, and strictly follow the Placement Docs updated in the facebook group. Companies tend to repeat questions across campus tests, hence can be great help if the very same question pops up in your test too.
- Do not copy code from your friends or directly from other websites like Geeks for Geeks. Some companies have a very strict policy against plagiarism, and they will outright reject you if they find your code plagiarized.
- Whether you can use STL or not in the coding round varies between company to company. 90% of the companies allow you to use STL, but compilers provided by some other do not allow you to do so. (In our year, Microsoft and Google were among those.) Hence, before placement tests begin, go back to the basics, and understand how to take input from console, difference between C-style and C++ strings or how to use pointers (and not references!).
- Sometimes, during the test, you will write your most optimal solution to the problem, but still all test cases won't pass. Maybe the company expects a better solution from you or maybe you are on the edge: one extra unnecessary loop might be the cause of the 'Time Limit Exceeded' error. In such a case, sometimes writing code in a different language works (e.g. Python). Generally, the Time Limit considerations are built in differently for each language, hence if you are lucky, you will pass over that 'edge' by simply writing it in a different language. (Note that, this should be your last resort, when you know, there can't be a more optimal solution to this question.)

Diversify

Diversify! Do not put your entire focus on one company or one profile only. When you have gone through CDC placement process, you will realize that the process is not entirely fair. It has a significant amount of *randomness* included. Some people may get placed in a better company than they expected, while some may not even get shortlisted in the company they prepared for last 4 months. It happens. There are tons of applicants, and only a few has to get in. Even if you are among the best, it may happen that you didn't make the cut. So, what is the best way to beat the *randomness* factor?

Diversify. Have a dream company in your mind, but prepare well for other companies too. Do not stick to just one profile. If you are preparing for Data-Science primarily, do some coding on the side too.

CDC Placement is not end of the world

Consider your CDC Placements as a chance to start your career well, and not as the most important part of your career. This is not JEE. CDC Placements do not have even 1% of impact on your future career compared to what JEE had. This is your first job, first of many more to come. You will definitely want to have a good start to your career, but understand that, if you do not get placed in your dream company or your dream profile, you can always make a switch sometime soon, if you work hard enough. Do not take your campus placement to heart. In 10 years, it won't matter.

FAQs

1. Does Department matter ?

Unfortunately, yes. It matters quite a lot, especially for software profile companies coming in Day 1/2. The difference arises in 2 ways: one, companies opening for CS/ECE/EE/Maths are quite more than other branches, and two, even if companies open for other branches, companies have a bias towards CS/ECE/EE/Maths students while shortlisting them for interviews.

Another difference is, companies have a higher CGPA cutoff to let candidates appear in tests, for other branches, compared to CS/ECE/EE/Maths. For example, a certain company X, but might have a CGPA cutoff of 7.0 for CS/ECE/EE/Maths, but 8.0 for other branches.

2. Does CGPA matter ?

Yes. Having a higher CGPA is always a plus, both while getting shortlisted for interviews and clearing them. Although, you don't necessarily need to have a CGPA of 9.5+, but having a CGPA of 8.5 gives you the ticket to appearing in the tests of all the companies that open for your department.

If you are from CS/ECE/EE/Maths, but have a CGPA <8, you may miss out on some companies, like Samsung Research, Paypal, Apple, Cohesity, Nutanix etc. CGPA also matters if you are from non-circuit branches, as discussed above. These branches might have a higher CGPA criteria to appear in the company's tests.

3. Does AIR matter ?

Well, this is a funny question, but 99% of the companies don't ask about your All India Rank in JEE. The other 1% includes trading firms like Quadeye.

Resources

Coding

Websites:

1. Geeks For Geeks
2. Interview Bit
3. LeetCode

Books:

1. Cracking the Coding interview (Must read)

Computer-Science:

1. Operating Systems YT playlist
2. System Design YT playlist, by Gaurav Sen
3. Gate CS Topicwise Notes - GeeksForGeeks

Other resources to checkout:

1. LeetCode Weekly contests
2. LeetCode questions arranged in order of likes
3. Dynamic Programming YT playlist, by Tushar Roy
4. Geeks for Geeks Quizzes
5. Operating System notes, by Anubhav Jain
6. Some interesting Algo/DS questions
7. CDC Preparation blogs (Only accessible via campus network)

Quant/ Prob-Stats/ Puzzles

Books:

1. 50 Challenging Problems in Probability (Must solve)
2. Heard on the Street (Must solve)
3. How to Ace the Brainteaser Interview

Practice basic maths and aptitude questions:

1. Pariksha
2. Prepleaf
3. Brilliant

Puzzles:

1. Puzzles from Geeks for Geeks
2. Puzzles from Interview Bit
3. CSE Blog
4. Gurmeet
5. The Grey Labyrinth
6. Nigel Coldwell

Other resources to check out:

1. Probability and Statistics Notes, by Anubhav Jain
2. Collection of some advance Dice problems

Machine Learning/ Data Science

Books:

1. Introduction to Linear Regression analysis
2. Probability and Statistics in Engineering, by Hines and Montgomery

Websites/ Blogs:

1. ML Resources
2. CS 229 Topic-wise notes, by Andrew NG
3. Interesting Notes and Explanations
4. Colah's blog

Courses:

1. Machine Learning, by Andrew NG - Coursera
2. DeepLearning.ai
3. CS229: Machine Learning - Stanford (Advance)
4. CS231n: Convolutional Neural Networks for Visual Recognition (Advance)
5. CS224n: Natural Language Processing with Deep Learning (Advance)

Miscellaneous

1. Inter IIT/NIT/IIIT/BITS Placement 2021 Facebook group
2. Collection of some of above mentioned books