Your phone interview will take an hour and you will need to have a computer with internet access for the coding portion of the interview, done through [www.collabedit.com](http://www.collabedit.com/).

To breakdown that hour for you in more depth, the first 5-10 minutes will focus on resume-related questions and introductions. The majority of the time will focus on working through at least two coding problems though. These questions will be heavier on your computer science fundamental/academic topics, touching on things like data structures and algorithms, hash tables and maps, big O notation, binary searches, recursion, etc.

Our engineers will be looking for your ability to not only code solutions for these problems, but also looking at your ability to work towards clean code and generate optimal solutions (i.e. are you looking out for edge cases, corner cases, bugs, are you able to generate test cases if asked), as well as your problem solving abilities and communication skills. From a problem-solving standpoint, are you able to demonstrate a strong logic process, can you think your way through to a solution, etc. In terms of communication, can you verbalize or discuss your methodologies and approach with our engineers, can you ask questions in regards to clarify the problem at hand, etc.

[Please note the information below is geared to help you with preparing in general for LinkedIn and not specific to any position. There will be additional questions specific to your area of interest.]

Major components of our assessment involve:

- Applying algorithm, data structures and Computer Science fundamentals

- Explanation and implementation of the solution in code without any of the aids engineers typically have access to (e.g. IDE, online docs).

- Pace-- you will be under the time gun.

Algorithms hints:

-  Think about and verbalize the trade-offs of different approaches.

-  Knowing the run-time complexity of your solution may indicate whether you can do better.

-  Don't over-complicate your algorithm to fit in specific concepts - we are looking for answers to the question, not looking for certain techniques.

Coding sites to visit for practice questions:

[Leet Code](https://leetcode.com/)

[Top Coder](http://www.topcoder.com/tc?d1=tutorials&d2=alg_index&module=Static)

[Developers book (Core Java)](http://www.developersbook.com/corejava/interview-questions/corejava-interview-questions-faqs.php)

[Stack Overflow](http://stackoverflow.com/questions/58354/algorithm-data-structure-design-interview-questions)

[Dynamic Programming Fundamentals](https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/)

http://courses.csail.mit.edu/iap/interview/materials.php

I know interviewing can be a stressful process, so here are a few additional tips to help you:

Tip #1: ***Ask*** clarifying questions instead of making assumptions. Before starting on your implementation, be clear on requirements. Describe your high level approach first before implementing.

Tip #2: ***Verbalize***your thoughts out loud while you’re solving a problem. Our interviewer wants to understand your thought process and how you go about solving a problem. So by thinking out loud, you’ll give them an idea and they’ll be able to give you hints to steer you in the right direction. Clear technical communication is key. Give your interviewer an idea of what they should expect before jumping into the solution. Take notice of suggested hints.

Tip #3: You will be assessed on both ***speed*** as well as ***quality***of the solution. So, don’t take too much time to think through the problem, but take enough time so you can provide a clean and elegant solution. For some questions, the expectation is that someone should be able to work through the problem in “X” minutes, so there could be multiple questions in the hour slot; others may take the entire hour to solve.

Tip #4: If asked about an interesting project, make sure to choose a project that you are actually interested in rather than just picking your most current project. Show ***enthusiasm*** *and* ***details*** about the project by explaining who was your customer, why the project/product was being built, and give detailed information about your contributions and specific responsibilities.

Tip #5: ***Find and fix*** your bugs using edge cases, tests, or other means.

Tip #6: ***Visit*** our engineering blog to find out a bit more about the different technologies and projects we are working on as well as get a feel for our engineering and company culture. Here are some helpful links to learning more about LinkedIn Engineering:

[Engineering Blog](http://engineering.linkedin.com/)

[Data @ LinkedIn](https://engineering.linkedin.com/data)

[Open Source @ LinkedIn](https://engineering.linkedin.com/open-source)

[LinkedIn's Culture](http://blog.linkedin.com/2015/03/11/linkedins-culture-of-transformation/)

[Engineering Insights (Video interviews with engineers at LinkedIn)](http://www.slideshare.net/linkedin/tagged/LinkedIn%20Engineering%20Insights)

Tip #7: Review your ***data structures*** and the libraries for them in your language(s) of choice, particularly paying attention to their limitations.

Tip #8: ***Ask*** relevant and appropriate questions about LinkedIn to show your interest. This is your chance to ***interview us*** to make sure that LinkedIn is the right fit for you.

Lastly, below are links to posts written by our engineers. They give a lot of info and provide different perspectives on how to prepare for this type of interview:

[What to do (and not do) if I interview you](https://www.linkedin.com/pulse/20140425202654-10752460-what-to-do-and-not-do-if-i-interview-you?trk=mp-reader-card)

[Interviews are not Exams](https://www.linkedin.com/pulse/interviews-exams-jacob-kessler)

[Getting That Job At LinkedIn](https://www.linkedin.com/pulse/20140205163949-28614372-getting-that-job-at-linkedin?goback=%2Enmp_*1_*1_*1_*1_*1_*1_*1_*1_*1_*1_*1&trk=object-title) (I strongly recommend that you spend the most time reviewing this article.)

[Technical Design Interview](https://www.linkedin.com/pulse/technical-design-interview-guide-success-joey-addona?trk=mp-reader-card)

[Success in the Coding Interview](https://www.linkedin.com/pulse/success-coding-interview-joey-addona)

[Design for Scale #1](http://www.lecloud.net/tagged/scalability)

[Scalable Architectures](http://tutorials.jenkov.com/software-architecture/index.html)