感想：

* 工作五年，面的level都是senior，一般onsite就是一轮design，或者有些coding最后有时间也会问一个小design。感觉design的表现对hire/no hire的影响越来越大，同时也直接决定了你的level和工资，因为coding只要花足够时间准备了大部分都没啥问题。。。就像fb的coding轮只有过和不过，design轮才决定你的level
* 遇到的design可以分三种：general design (design facebook), real work design (面试官拿一个他工作中真正遇到的问题，一般比较具体) 和 domain knowledge design (你做的东西和职位非常match，比如database internal或者AI)
* 我做的东西比较专，所以domain knowledge还不错，但是general design的话基本都没有hands-on experience, 大部分知识都是通过看资料学习的（我连REST API都没有写过），虽然准备了好久结果也不是很好，到最后的评价还是广度够而深度不够。如果你和我有一样的问题，我其实也没什么好方法，design轮fail的飞起，越fail越觉得自己不行。。。但是有几点我觉得比较重要：

               o        随着花时间看各种各样的资料和实战，我可以感觉到我的design水平（不管是硬实力还是套路）肯定是在一点一点提高的，一个知识点可能在好多资料里重复出现（比如message queue），我的理解也会越来越深，所以请一定不要放弃  
               o        一个人基本不可能所有component都做过，所以要尽量把面试官往自己熟悉的方向引导，回答最好要有亮点（面试官都没有想到的点）  
               o        solution多提几个option，就算很明显有问题的也可以提，之后讨论trade off，最后选择一个并解释为什。完了你每加一个东西都要解释一下为什么加它，是不是必要的  
               o        设计可以从简到繁，不然可能会被说over design。。。  
               o        如果到最后你觉得光看资料的帮助已经很小很小了，还有一个option就是真的去做一些小的project来增加一下hands-on experience（本人没有试过，听朋友推荐的）

* 我的流程一般是（这个只适用于没有经验的朋友，高手可以具体问题具体分析）：

               o        Functional requirements  
               o        Nonfunctional requirements （availability, consistency, heavy read/write, etc.）可以顺便提一句需要monitoring system(technical/business metrics)  
               o        Scale estimation (DAU, QPS (low latency), etc.)  
               o        API design + database schema  
               o        Overview design workflow  
               o        What are the failure cases and how the system could handle it  
               o        Scale up/out  
               o        Deep dive in one area (需要和面试官check)

* Mock的话就是找朋友，免费网站试过Pramp和interviewing.io一两次，感觉都还可以

资料：

* designing-data-intensive-applications （书）
* web scalability for startup engineers （书）
* https://www.[educative](https://link.1point3acres.com/?url=http%3A%2F%2Fwww.educative.io%2Fexplore%3Faffiliate_id%3D5749180081373184%2F" \t "_blank).io/courses/grokking-the-system-design-interview/ (地理好像有折扣码）
* https://www.interviewbit.com/courses/system-design/topics/interview-questions/
* 九章算法系统设计
* https://github.com/donnemartin/system-design-primer
* Medium和各个公司的engineering blog
* https://github.com/puncsky/system-design-and-architecture
* https://www.1point3acres.com/bbs/forum.php?mod=viewthread&tid=552608&extra=page%3D1%26filter%3Dlastpost%26orderby%3Dlastpost%26orderby%3Dlastpost
* http://highscalability.com/all-time-favorites/
* https://time.geekbang.org/column/article/132434
* https://github.com/checkcheckzz/system-design-interview#tips
* https://soul[mac](https://link.1point3acres.com/?url=https%3A%2F%2Fwww.amazon.com%2FApple-MacBook-16-Inch-Storage-2-3GHz%2Fdp%2FB081FV1Y57)hine.gitbooks.io/system-design/content/cn/

在职跳槽已经结束了一段时间，还是想把自己一路下来的感想和经验分享给大家，过程之艰辛只有自己知道，不过大部分人都要走这一步，没经历过[刷题](http://www.1point3acres.com/bbs/forum-84-1.html" \t "_blank)和面试过程的痛苦，又怎么能感受到拿到dream offer那刻的喜悦呢？  
Note: 这篇分享基本没有关于如何拿到面试机会的东西， 因为本人大厂工作五年，所以拿到面试机会对我来说不是什么问题（疫情期间不招人的公司除外）。  
  
简历：  
工作时间比较长所以可写的东西还是有一些，两点我觉得非常重要，第一点是要简明扼要，不要写满满的一页而没有重点，面试官找的也是关键字；第二点就是要有数字，比如improve perf by X%, reduce cost by Y%.  
  
HR面：  
Phone interview之前一般都会有一轮HR面10-30min，就是对你情况的基本了解，问一些问题像为什么离开现在公司？目前面试进度？注意的点就是一定要表现出对公司的兴趣，我在聊之前至少都会花一点时间了解这个公司。  
  
Behavior：

* 问题一般分两种，一种是general的，比如most challenge project，可以发挥的方向很多。还有一种就是非常具体的问题，问你在某种情况下最好的解决方法，比较难准备，所以平时不要埋头coding，多看看厉害的senior是如何待人处事的。
* 叙述故事的时候最好用STAR模式，一定要挑你有high impact的说，就算重复也没关系，不然会被说impact不够。描述的方式也很重要，就算实际上很简单也要说的给人一种很复杂的感觉。。。 网上可以找到很多具体问题，举几个常见的例子：

          o Most challenge/proud project  
          o Failures  
          o Miss deadline  
          o Conflict with manager/colleague  
          o Negative feedback

* 推荐看一下[Amazon](https://link.1point3acres.com/?url=https%3A%2F%2Fwww.amazon.com%2F" \t "_blank)的14条军规([https://www.amazon.jobs/en/principles](https://link.1point3acres.com/?url=https%3A%2F%2Fwww.amazon.jobs%2Fen%2Fprinciples)), 说的点就可以往这些上面套
* **Resume上写的project**肯定都要过一遍，不要被问住
* 控制时间很重要，比如phone的话就不需要说得太具体，不然coding时间会不够，而专门的behavior或者deep dive轮就可以说得很详细。而且要根据具体情况决定要不要说很多technical的细节。
* 剩下的就看个人魅力和平时积累吧，毕竟当过tech lead或者大项目的project lead是肯定可以说出来很多东西的

Coding：

* 刷题效率很重要，拖得时间越长之前做过的就忘记了。。。所以**第一次做5min想不到答案就直接看，不要浪费时间**
* 刚开始先按类型刷，每个类型先做五道，可以网上搜索或者用LC自带分类，推荐有空看看youtube花花酱的视频
* 每道题都要吃透，理解所有方法和time complexcity。犯的很蠢的错误一定要记下来，过一阵子反过来再做一遍
* 面试的时候见过的题不要直接最优解（有些公司不适用），一步一步解释你是如何得到这个最优解的，这样更能体现你的能力
* 没见过的题也不要一直纠结给出的答案是不是最优解，**写完并且没错误是最重要的**
* 高频题大家都会做，这时候拼的就是细节了，object naming，代码整洁之类的就比较重要了，所以平时练习的时候也要多注意这些
* 网上有很多mock interview的网站，pramp我用过一两次还可以
* 如果你fail还因为coding题做不出来的话，我觉得这块是比较好提高的（相对于design），多做题多总结就完了，200道不行就1000道，1遍不行就5遍，干就完事了～

Design:

* 之前写的<https://www.1point3acres.com/bbs/thread-644973-1-1.html>

面试安排:

* Phone的话疫情期间安排比较随意，看你什么时候状态最好。onsite一周最多两个，因为需要时间**过一下面经题**，不然我总感觉没准备好。。。当然我看也有很多人一周3-4 onsite的，主要是想把offer放到同一时间好compete。。。我个人觉得是没必要的这么累的，你只要跟公司说自己还有几家想面，一般公司只要没出纸质offer的话都能等一个月左右（个人经历，不一定所有公司都这样哈）
* 面试公司的数量和顺序根据你具体情况而定，我的建议是根据自己喜好把公司分成low，medium和high，high是最想去的几家公司，先从low/medium里面选几家开始练手，什么时候觉得准备好了再开始面high。[https://www.1point3acres.com/bbs/thread-631582-1-1.html](https://www.1point3acres.com/bbs/thread-631582-1-1.html" \t "_blank)里面有我总结的西雅图地区所有公司，供大家参考。
* 大公司从hr联系到onsite每一步都要至少两个星期，所以安排不好就可能面试之间隔很久，所以我的建议是大公司先走流程，最后如果没准备好面试时间往后拖一拖也没关系

Negotiate offer:

* 这块我没有花很多时间研究，凭借运气和实力拿到了自己最想去公司的offer，所以当时想的就是只要不跟别的offer差太多就去了。最后compete了两轮，**第一轮是用其他公司offer，第二轮就是你能给到多少多少我就马上签offer**。现在想想有一点小后悔的是最想去的公司没有太敢多要，主要是因为网上data point太少+怂了，怕要太多会被直接拒绝。现在想想还是就应该要求match最高offer的，就算没成功也有可能加到最高限度，最坏的结果就是被直接拒绝，说明没缘分，那就去offer最高的呗。。。
* 查package我用的就是**blind**，地里和**levels.fyi**