

作者: Windson Yang

本文章由[ResumeJob](#)贡献, ResumeJob能帮助你重新规划你的职业生涯, 商业转载请联系作者获得授权, 非商业转载请注明出处。

我刚开始工作的时候, 对面试总是很恐惧, 一方面怕简历无法通过筛选, 另一方面怕现场发挥得不好, 浪费自己和面试官的时间。**我当初并不知道对比其他求职者, 自己的优势和劣势在哪里, 也不知道自己到底值多少钱。**面试的时候都觉得面试官很严肃, 问的问题我也都答不上, 肯定没有机会入职, 觉得自己烂透了, 也不想浪费时间去面试多几家了。

现在回过头看, 我觉得当初自己在欺骗自己, **我真正担心的是即使自己认真准备简历和面试也对结果毫无影响, 更不敢去想如果找不到工作没钱缴房租怎么办, 索性海投简历碰运气。**其实认真准备简历和面试是很有用的, 因为毫无准备, 随便面试的求职者真的太多了。工作了几年之后, 当我有了面试官的经历之后。我慢慢发现, 公司招人实在困难, 招的人要有计算机基础知识, 有项目经验, 愿意学习而且愿意来这家公司, 靠的完全是缘分。所以只要你有一定的实力, 愿意努力, 现在的互联网行业还是买方市场, 认真准备总是有收获的。我希望把这几年作为面试者和面试官身份的经验给大家, 也大家可以学到从中一些面试的技巧, 找到心仪的工作。你可以使用[ResumeJob](#)进行简历Review和模拟面试, 这样真实面试的时候更有信心。

- [分析阶段](#)
- [准备阶段](#)
- [面试阶段](#)
- [总结阶段](#)

[分析阶段](#)

- [公司需要怎样的员工](#)
- [员工需要怎样的公司](#)
- [了解自己的优势](#)

公司需要怎样的员工

先从公司的角度出发, 如果你是部门的总监, 你会希望什么样的员工?

我很喜欢Google前CEO施密特分享的一个故事，他刚到Google的时候，Google还只是个小的创业公司。他以为Google和其他公司没什么两样，直到有一个周五，拉里佩奇在用谷歌搜索一些关键字的时候，他发现出现了一些不相干的广告推荐（这个情况我们现在在国内最大的搜索引擎也经常看到）。施密特以为接下来就是开几个会议，然后分到具体的工程师手上解决。但是拉里佩奇没有这么做，他用纸条写下"These Ads Suck!"，附上相关的截图贴在布告栏上就回家了。接下来的72小时彻底改变了施密特的认知。**在周一凌晨5点，有几位并不是负责广告业务的工程师发来一份邮件，从头到尾阐述了这个问题产生的原因，他们的解决方案，以及这个计划对公司有什么影响。这和我们强制的996加班不一样，他们从公司的角度去思考，自愿自发地用周末的时间去解决并不属于自己范畴的问题。**这样的员工我想就是每个公司都需要的：

- 有技术，工作认真负责，可以及时解决问题，能给公司带来效益
- 出色的团队合作精神，能与团队一起成长
- 愿意自我学习，投资自己

所以求职者的简历，面试中必须能体现出这几点品质。举个例子，要展现自己喜欢学习计算机，与其说：

热爱计算机，喜欢学习计算机系统的知识。

可以修改为：

喜欢阅读计算机系统的书籍，完成《深入理解计算机系统》80%以上的习题。并在博客分享学到的知识。

团队合作能力也是面试官非常注重的一点，面试过程中如果被问到有没有带领团队的经验，如果你没有的话也不要直接回答没有，可以根据自己的情况回答：

“我在以往的项目中与团队成员都能融洽相处，并且每个月都会做定期的技术分享互相学习，虽然没有带团队的机会，但是相信自己能够做好。”

当然这些不是一下子能想到，面试技巧多练就好。**定期找几家公司面试练练手，一方面能知道市场的行情，另一方面随时能找到其他更好工作机会，跳出现有的舒服区。**

员工需要怎样的公司

找新工作之前，先认真思考下几个问题，一份工作中你最看重的是哪些方面？

- 薪酬
- 公司福利 / 工作环境
- 工作方向（假如你要从技术转向管理，这个岗位提供这样的可能吗？）
- 公司发展前景
- 个人成长

了解自己的想法之后，再进行筛选公司以及准备如何提问。有时候薪酬的远不及公司文化重要，如果公司有好的工程师文化，能聚集一批优秀的工程师。那么只要认真待着一两年，进步速度要比其他地方快得多。**你可以根据自己的着重点，筛选不同的公司，以及准备向面试官提问的问题。而且非常重要的一点是，也可以真正地去思考你到底喜不喜欢这家公司，面试和找男朋友一样，找你喜欢的，而不是找你能找到的。** 这点我们在[面试阶段](#)会介绍，提问这部分非常重要，我遇见很多求职者感觉不好意思提问，一方面怕面试官会觉得自己有很多要求，另一方面根本没有想好要问什么。这点我觉得非常不明智，公司和员工就是互相选择的，一定要多提问题，了解公司的文化以及岗位的职责。才不至于刚入职就因为不适应而要离开。**如果求职者没有提问，面试官也会担心求职者入职后会不会不适应公司的某些规定，例如加班规定，着装要求。由于解雇一位员工比招聘要痛苦得多，面试官反而会不敢轻易录用他。**

了解自己的优势

仔细分析自己的优势和劣势是什么，然后添加在简历中，优势可以从这几点入手，后面是面试官的理解：

优势

- **大型项目的经验**（能解决项目普遍出现的问题，代码能力应该不会差）
- **作为主要参与者得过奖项**（聪明，勇于尝试）
- **维护优秀的开源项目**（懂得团队协作，能写出优秀的代码）
- **发表过论文或者优秀的博客文章**（研究能力强，分析能力强）

- **数据结构和算法基础好，Leetcode中等难度都能bug free**（基础不错，培养起来也简单）
- **思维敏捷，聪明，有想法**（需要有文章，论文，得奖来证明）
- **毕业于不错的学校**（学习努力）

与上面相反的就是劣势了。了解到自己的优势和劣势，简历就可以强调自己的优势，面试根据自己的劣势来准备。例如，如果你要面试的是一家公司的开发工程师，那么项目经验以及对框架的熟悉程度就比较重要，如果面试的是研究岗位，那么论文与文章的数量就很重要。

准备阶段

- [随时都在准备](#)
- [技术知识](#)
- [积累项目经验](#)
- [准备简历](#)
- [模拟面试](#)
- [选择公司](#)

随时都在准备

随时准备并不是鼓励频繁地跳槽，而是要有随时有跳出舒服圈的准备，也许你已经很满意现在的工作，薪酬，觉得习惯而且安逸。不过如果公司突然倒闭，或者部门被裁减，你还能找到这样或者更好的工作吗？**我建议各位求职者，每三个月可以去面试一两家公司，因为你已经有不错的工作了，所以可以带着轻松的心态去面试，每一次面试都是很好的磨练机会。**面试就像打篮球投球一样，多练习才会准，如果一直害怕不进，连投球都不投的话，那连碰运气的机会也没有了。我刚参加工作的时候是一份烂透的简历---普通大学，专业不是计算机，作品也没多少。也有一间大公司给我面试的机会。（我没有做好准备，所以第一次的时候失败了，第二次成功入职了。后来我知道互联网行业招人真的非常难。）。机会随时都会来，如果明天你喜欢的公司让你去面试，你准备好了吗？当你的GitHub看起来不错的时候，很多公司的HR就会邮件联系你。你有信心去面试吗？工作空闲之余看看优秀的博客，研究下算法题，日积月累下来知识与计算机基础就会越来越牢固，不用面试的时候临时抱佛脚了。

技术知识

计算机科学基础

- 操作系统基础
- 网络协议基础
- 数据结构与算法基础
- 堆栈，链表，队列，二叉树，哈希表，图以及对应常用的操作方法
- 二分法，位操作，递归，动态规划
- 七大排序算法

系统设计

- 缓存，分布式存储/计算
- 设计模式
- 数据库基础

我推荐几个学习资源：

- [Cracking the Coding Interview](#)
- [tech-interview-handbook](#)
- [Leetcode](#)
- [算法入门](#)
- [可计算性、复杂性和算法](#)

刷算法题的时候，要把每道题都当成面试题一样按步骤完成，完成一题之后总结经验。这样遇到变形题也迎刃而解。可能有的同学有疑问，觉得这些平常工作都用不到，为什么还要花那么多时间在上面。其实不是的，第一，平常工作都能用到，无论从二分查找到复杂一点的贝叶斯模型。开发的过程中如果你知道这个算法 / 数据结构，就能根据自己的业务来选择最适合的算法 / 数据结构，减少项目的复杂度。第二，数据结构和算法锻炼的是思维，当你刷算法题例如leetcode的时候，慢慢会学习到一些有趣的，巧妙的方法。它们能扩展你的编程时思考的范围。同时也要求你考虑到各种不同的边界情况。**即使你不准备换工作，我也建议每天都刷一道算法题，日积月累，一年下来你的算法基础一定能比同龄人高出不少。而且当你真正理解算法题的知识之后，写程序debug和花在stackoverflow的时间就会大大减少，往往知道哪里可能有问题并且能大幅地增加工作效率。**

积累项目经验

- 参与开源项目

计算机科学最重要的就是实际运用，关键就是多参与，多思考。开源项目可以让你和世界上顶级的工程师一起工作，学习软件设计以及语言的高级使用方法。同时能让你理解软件的是如何运行的。

- 参与较底层 / 较复杂 / 偏向算法或研究的项目

如果你未来想从开发转向研究的岗位，那么就可以阅读一些相关学术论文，写相关的文章分析与工具。

- 造轮子，写一些实用的工具

从学习的角度来说，造轮子可以说是最好的方法，不过你要给自己一个期限，不能无止境地把时间花费在程序的细节中。知道原理，能够实现其实就非常好。尝试从实现平时常用的[Web服务器](#)，Web框架开始，有兴趣可以延展到操作系统或者编程语言（我遇到过这样的求职者）。自己写完再看看别人是如何实现的，学习他的优点。其实到最后，你会发现计算机是越学越容易的，如果你不了解同步异步，往往是因为你不知道Web服务器是如何实现，不知道系统调用是如何实现的。当你能自己去实现的时候，很多以前的问题也就迎刃而解了。

- 写论文，分享文章

如何宣传你的开源项目或者业余项目？写一篇优秀的文章介绍它。同理，要证明你有喜欢计算机，有研究的能力，最好的方法也是写一些优秀的文章以及论文。

准备简历

- [找出优势](#)
- [给出证明](#)
- [其他能力](#)

我筛选过超过千份的简历，遇到太多太多千篇一律的简历了，描述都是熟练精通xxx框架，项目经验都是博客加爬虫。无论你是计算机专业，还是其他专业，还是培训班出来，都应该

重视自己的简历。要知道哪些简历是“不好的简历”，这里指的不好，未必是求职者个人的问题，而是他没有从面试官的角度来思考编辑简历。几个要点是要注意的：

找出优势

简历并不是越长越好，最好的简历长度是一到两页，列出你最优秀的项目经验以及奖项。至于语言或者框架，只是简单接触过的话就不用写上去了。面试官问你有没有学过其他的时候才说出来。**假如你只是学过简单接触过Go，面试官会先入为主地以为你对这个语言比较了解，如果问一些基础的问题你也没有回答正确。这样反而会给面试官留下不好的印象，因为他会认为你对简历中的其他你真正熟练的语言也不太了解。**常看到的写法是：

精通django框架，熟悉Python语言

建议修改为：

精通django框架，是django的Top50代码贡献者。熟悉Python语言，理解Python垃圾回收，迭代器，装饰器等常用对象的实现原理

多花几分钟的时间，就能在求职者的简历里面脱颖而出了。

给出证明

常规写法：

前期负责前后端API设计，后期负责实时流消息处理应用系统构建和实现

面试官无法知道你做得怎么样，建议根据“发生什么事”，“你做了什么”，“结果怎么样”三个点来修改，同时这里必须出现数据作为参考，例如：

推动团队转用Graphql为新的API接口规范，从而减少20%的日均请求量，并节省了两台服务器资源。后期负责实现使用RTSP协议进行实时流消息处理，经过测试与优化，接口请求响应时间平均为40ms，同时架构了能支持100万日活量的缓存服务器与后台服务器。

这着并不是吹嘘，因为你在项目中做的每一个选择必然是有原因的，而且必然会对项目产生影响。而你在项目经验中要写的就是把你做的最优秀的项目中你产生最大的影响那部分。**如果只是想面试官到时候有兴趣的时候问再答吧**，到时候如果紧张，很容易忘记具体的数字。如果项目经验不多，可以把学校的专业排名(50/1000)，员工考评(10/1000)，优秀员工这些指标都加上去。**不要觉得没有用，这绝对是大多数求职者忽视但是非常重要的点。**而且简历如果既有一些较新的技术（例如Rust，Go，当然你要真的了解），又有经典的必备的技能，那么就一定能够吸引他的眼球。

其他能力

其他能力就是团队协作能力以及解决问题的能力，如果你已经在开源项目有不错的贡献，那么面试官就不用担心团队协作能力。至于解决问题的能力，你可以在项目经验中可以列出解决的比较复杂的问题，例如“解决了服务端同时推送10万台设备的的并发与资源占用过多问题”。这样面试官就知道你既有团队协作能力又有解决难题的能力。面试题就不会出那么难了。其实很多公司在面试的时候都会出一些非常难的题，并不要求面试者一定要解决，而是要看面试者在遇到难题的时候会怎么面对，是思考一下就放弃，是寻求面试官提示，还是从多个角度去解决问题。如果在简历中已经体现了这一点，那么面试的时候就能略微放松了。

模拟面试

这个大家可能接触得比较少，如果你准备去面试一家非常喜欢的公司，面试之前，你应该先进行模拟面试，模拟面试既可以让你的朋友当面试官来面试你，也可以去找几个有类似岗位的公司。第一，当你本来就没有一定要进模拟公司的想法，那么心态自然就能放轻松，带着轻松的心态去面试的话更能发挥好，给自己信心，同时也可以问问面试官自己哪里不足，可以加强的。同时锻炼自己面试的技巧，包括技术的基础，以及如何问问题。当你面试得多了，发现其实都差不多，遇到问题也有经验了。平时也可以在一些在线网站上面模拟，例如为外企求职使用的[Pramp](#)。

选择公司

前几家公司的选择对你的职业规划会有很大的影响，大公司还是小公司，和你个人的性格或者职业规划有直接的关系：

大公司

- 优势
 - 入职薪酬比较高，每年固定调薪，员工福利例如下午茶，文娱活动，年假都有保证。
 - 通常都有大牛，而且工程师比较多，总能遇到一些志同道合的朋友。
 - 通常加班压力没那么大。完成份内工作即好，开发工程师不需要部署和测试。
 - 跳槽到小公司比较容易，岗位也能得到提升。
- 劣势
 - 刚进去的时候接手的可能都是比较枯燥的小项目。
 - 可能需要维护几年前的没有文档没有测试的项目。（其实也能学习到很多）
 - 比较容易安逸，缺乏学习的动力。

小公司

- 优势
 - 相对来说，条条框框没那么多，偶尔迟到请假没什么关系。
 - 什么都能学到，从开发到运维到测试。
 - 项目可以加上自己的建议和想法，比较有成就感。
 - 能直接向老板汇报，升职速度比较快。
 - 万一上市了呢？（想想就好）
- 劣势

- 入职薪酬比较低，员工福利嘛，不能保证。
- 有大牛的可能比较少，除非你事先知道（所以面试问问题非常重要）。
- 加班压力通常比较大，而且公司不一定会根据你的加班时间就涨薪。
- 跳槽到大公司比较难，除非你有非常好的简历与能力。

在国内来说，小公司并没有那么自由，加班也可能很多。所有我觉得一开始选择大公司往往是不错的选择，之后跳槽的选择范围也更多。

面试阶段

- [HR电话确认](#)
- [电话面试](#)
- [家庭作业](#)
- [现场面试](#)
- [非技术问题](#)
- [求职者提问](#)

当你得到了面试的机会，开始进入重头戏了，无论你的履历如何出众，都不能对面试掉以轻心。我遇过很多简历不错但是面试一塌糊涂的求职者，结果当然没有录用他们。翻转二叉树起码要会写吧 :D 面试一般会有几轮：

HR电话确认

HR会和你聊下天，确保你了解这个岗位的基本信息。也可能问几个关于你简历的问题，这轮只是考核下你的基础信息是否正确，看看你的谈吐是否正常（相信我，很多求职者如果不看自己的简历，连自我介绍都做不到）这轮放轻松，实话实话就好。

电话面试

这是技术面试的第一轮，可能会电话提问，也可能是通过把题目发在在线文档，然后让你去解决。一般都是比较简单的算法，数据结构的基础问题。如果遇到难的也不需要担心，提供

一些解题的思路，即使最后不能bugfree，起码也有一些头绪。

家庭作业

这轮并不常见，有的公司会让你实现一个小模块或者小工具。主要考核你实际情况下的开发能力。这点就要靠平时积累了，如何设计API，使用什么设计模式，都有讲究。多看开源项目源码就好。Python的话我推荐看[Requests源码](#)

现场面试

- [提问](#)
- [测试用例](#)
- [思考](#)
- [阐述](#)
- [伪代码](#)
- [代码](#)
- [检查](#)

现场面试可能会有简单的面试题，然后面试官出几道算法题让你可能电脑实操，也可能写在白纸或者白板上。我明白很多求职者不喜欢白板面试，也觉得白板面试没什么意义。不过在我面试的求职者中，白板面试能力强的在实际工作中表现得也比较优异。**白板面试确实难，不但对于你，对于其他求职者也是。要是你能做到，别人做不到，你就能在众多求职者中突围而出。**

简述一下解算法题的几个步骤：

出个经典题目[Two Sum](#)：

Given an array of integers, return indices of the two numbers such that they add up to a specific target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

给出一个整数数组和一个目标数，返回两个索引值，它们对应的数组元素的和等于目标数，只有一个答案。

Example:

Given `nums = [2, 7, 11, 15]`, `target = 9`,

Because `nums[0] + nums[1] = 2 + 7 = 9`,
return `[0, 1]`.

提问

这阶段的提问非常重要，因为你要100%地清晰地了解题目，才能解决题目。**不要觉得提问得多显得愚蠢，提问得多代表你在思考**，没有问题我反而会担心求职者是不是之前做过这题，或者根本没有思路。

1. 这是一个有序数组吗？不是（注意这里有个小陷阱，虽然在Example中给出的是一个有序数组，但是实际题目并没说这是一个有序数组，所以要考虑无序以及为空等边界条件）
2. 数组可以包含负数吗？不可以
3. 如果数组为空或者只包含一个数字，是没有答案吗？对的
4. 时间复杂度和空间复杂度有限制吗？没有

题目就转变成

一个只包含正整数的无序数组，要求返回两个不同的数组索引值，它们对应的数组元素的和正好等于目标数，如果数组为空或者只包含一个数字的话没有答案，其他情况有且只有一个答案。

这样就能排除一些边界情况了。然后写测试用例

测试用例

```
target = 9
```

```
# 测试用例
```

```
[],
```

```
[1],
```

```
[2, 7],
```

```
[2, 5, 7],
```

```
[5, 4, 2],
```

空的，只有一个元素，正序，逆序，正常情况都写下来，面试官会对你考虑到那么多情况而加分。

思考

先想想会用什么数据结构，链表，哈希表，堆，栈，二叉树，哪个结构能解决这个问题？如果真的没有思路的话，**思考了之后，可以请面试官给点提示**，这个其实也是团队合作的一种表现，请求提示不一定面试官就觉得你能力不行。

阐述

边思考边向面试官说出你的思路，虽然你的思路可能比较乱，没什么章节。但是没关系。要大声肯定地说出自己的想法，同时可以向面试官提问。我面试过几位求职者，虽然他们没有顺利地完算法题，但是能一直说出自己的思路，给我留下不错的印象。就像我之前说的，有些难题，面试官并不是期望你都能答对，只不过想知道你遇到难题是如何解决的。所以阐述自己的想法是非常重要的。

伪代码

如果数组长度小于2，返回False

建立一个哈希表

遍历数组每一个元素：

如果目标值减去元素值在哈希表中

返回该索引与当前索引

否则把当前索引与值添加到哈希表中

时间复杂度为 $O(n)$ ，空间复杂度为 $O(n)$

代码

这题算简单：

```
class Solution(object):
    def twoSum(self, nums, target):
        # 如果数组长度少于2的话，无解
        if len(nums) <= 1:
            return False
        tem_dict = {}
        for i in range(len(nums)):
            # 检测这个元素是否曾经出现过
            if nums[i] in tem_dict:
                return [tem_dict[nums[i]], i]
            else:
                tem_dict[target - nums[i]] = i
```

检查

把测试用例带进去代码中检查，然后看看哪里可能会有问题，做出修改。

非技术问题

接下来面试官可能会问一些非技术的问题：

- Q: 为什么选择这家公司?
- A: 面试之前最起码要浏览过公司的网站, 了解公司有什么产品, 这样既可以防止遇到皮包公司, 或者小作坊欠薪拖薪。**也可以真正地去想你到底喜不喜欢这家公司, 面试和找男女朋友一样, 找你喜欢的, 而不是找你能找到的。** 回答可以说说你对这个公司的印象。
- Q: 你曾经面临最大的专业挑战是什么? 你是怎么战胜它的?
- A: 这个一定要准备好, 不能说没什么挑战, 没什么挑战代表你没有认真去思考, 就算是最简单的增删改查或者前端的动效, 背后的原理, 网络协议的原理, 你都应该去了解。对你在简历中的每一个项目, 你都应该能说出里面最大的挑战, 最有趣的部分是什么, 这样面试官才能真正理解你在项目中做了什么, 学习到什么。
- Q: 是什么为什么你选择离开你现任公司? 你从你上一家公司学到最重要的是什么?
- A: 大多数求职者不喜欢这个问题, 也不知道怎么回答。这个问题你能回答好的话就能拉出距离了。我觉得答案其实很简单, 你们公司的项目有更好的发展前景 / 我想挑战自己在这一方面的能力等等。至于说旧公司薪酬太低, 工时太长, 没前途这些就免了。
- Q: 你的长期工作目标是什么?
- A: 这个看个人, 转管理的话可能会加一轮问管理方面的问题, 转资深工程师的话可以讲下自己打算钻研哪个方向, 大数据, 人工智能, 区块链都可以。

求职者提问

这点非常重要, 要预防你到了新公司之后, 发现公司文化不适合你, 再马上找新工作的话就不好了。

- Q: 你们新老员工的比例是多少? 厉害的工程师有多少?
- A: 这个问题其实揭示了公司的文化, 如果新员工非常多, 公司也不算新的话, 那么代表流动率很高, 公司文化可能不是很好。第二个问题其实就是问有没有大牛, 有多少。有厉害的工程师总比没有要好得多, 进步的速度也更快。一个公司如果有比较多优秀的工程师的话, 代表是不错的公司。
- Q: 如果我入职的话, 会有入职培训吗? 会被分到哪个项目组, 项目组的成员构成是怎样?
- A: 这个可以了解公司的架构是不是清晰, 个人职责划分是否明确。如果面试官回答不了这个问题, 或者支支吾吾的话。即使你进去的话可能要兼顾几个项目, 维护

老项目。这些都要问清楚，你才知道自己大概的工作量有多少。维护旧项目虽然头疼，但是上线压力不大。如果新旧一起来，就要考虑自己是否适合这样的工作强度。

- Q: 我入职的前三个月，要完成什么工作来证明我的能力呢？
- A: 这个问题其实为下一个问题准备，如果我工作表现优秀的话，公司会不会有对应的奖励？
- Q: 多久进行一次调薪，工作绩效是如何计算的？是按项目收益，还是主管决定。
- A: 这个也是了解公司有没有实施奖励制度，通常回答准备中的都要留个心眼，可能一年都不会调薪。
- Q: 公司的五险一金是按什么比例缴的，是按最低标准还是可以自己缴纳更高比例
- A: 一般这个回问HR，如果小公司的话，也可以直接问工程师。五险一金看似没多少，但是每个月累积下来就很多了，这个需要和面试官确认。
- Q: 我今天面试的表现怎样，如果通过之后我还会经过多少轮，怎样的面试流程
- A: 首先可以了解自己的不足，积累经验。也可以开始为下一轮复试做准备

总结阶段

一次面试过来，可能筋疲力尽了。回想下自己哪里可以做得更好，简历哪里可以修改的。统计学告诉我们不要选择第一家面试的公司，多面试几家。不要欺骗自己，认真去思考每家的优点和缺点，和你的好朋友聊聊，寻求他们的建议。如果没有拿到offer也没关系，重复上面的步骤，继续努力。两年前我连想都不敢想到美国的大公司工作，而现在的我就是在为Google的面试做准备，就算我现在进不了Google又有什么关系呢？我还是在准备过程中学到很多知识。我很享受这段时间。相信自己，努力和汗水总会能得到回报的。