第1章

Chapter 1

简历、面试和Offer

整个招聘流程主要如下：申请某个公司你感兴趣的职位，投出你的简历。HR从简历库中筛选符合要求的简历，安排面试。面试主要分为电话面试和现场面试，如果面试表现优秀，HR会同你讨论待遇、福利、起始时间等具体信息。针对上述流程，让我们一一分析其中的关键环节。

1.1 简历

简历是求职的第一关，尽管简历不能决定最后的录取结果，但是一份结构清晰、内容充实且具有针对性的简历，可以给你带来宝贵的面试机会。本节将从格式、内容安排和描述技巧等方面介绍如何生成一份对HR有吸引力的简历。可以根据不同的职位描述适当更改简历的某些内容，但不建议准备太多份区别过大的简历，以免自己投递的时候产生错误。如果你真的需要这么多简历以应对不同职位，则应当考虑下自己的优势在哪些方面，适当减少求职目标。注意，下文的描述注重在北美求职的简历结构，其中大部分要点具有普适性，但是如果在国内求职，有些地方可能需要根据实际要求进行更改。

1.1.1 格式

除非你想面试用户体验或者设计相关的职位，否则简历的格式尽量以简介清晰为主。标题及名字等重要信息可以适当加粗或者增大字体，其他文字描述尽量统一字号。字体选定一种即可，不建议用不同的字体去突出不同的内容：不同的字体会显得版面杂乱，并且对排版造成困难。英文简历Calibri是比较适宜的字体，中文可以用宋体。简历可以用Word、Latex、Pages或者InDesign制作，但是最后务必导出成PDF，确保在不同的平台上，对方看到的格式一致。

1.1.2 内容安排

简历是一个提供信息、展示自己的平台。对于刚毕业、没有什么工作经验的人来说，简历一般控制在一页为宜，如果是PhD，需要适当列举一些相关的文章发表信息，那么简历可以扩展至两页。这里就给一个示例，它在一页纸的长度中精炼地描述了求职者的教育背景、实习经历、技能和专业背景，如图1-1所示。

简历的第一要务是让对方能够认识并联系上你，所以名字、邮件、电话、地址等基本信息缺一不可。特别的，对于在美国本地求职的人而言，邮编很重要，因为网上的申请系统往往会根据邮编将申请者按地域划分。对于很多公司而言，他们会优先考虑本地的申请者。所以当你申请的公司在当地有总部/分部时，务必写上本地的邮编。

Anakin Skywalker

CONTACT INFORMATION	School of Computer Science, Carnegie Mellon University 5000 Forbes Avenue Pittsburgh, PA 15213	Tel: (XXX) XXX-XXXX E-mail: xxx@andrew.cmu.edu Website: http://www.website.com

EDUCATION

Carnegie Mellon University, Pittsburgh, PA
Master of Science in Information Technology August 2013 – December 2014 (expected)
- Specialization: XXX, GPA: XXX
- Coursework: Multimedia Database and Data mining, Machine Learning with Large Datasets, Distributed Systems, Web Application Development, Operating Systems

XXX University, Beijing, China (PR)
Bachelor of Science September 2009 – July 2013
- Major: Computer Science

PROFESSIONAL EXPERIENCE

XXX Inc., Palo Alto, CA
Software Engineer Intern May 2012 – August 2012
- Benchmarked XXX performance by implementing a XXX with XXX.
- Profiled and explored ways to improve performance for XXX. Reduced the processing latencies by XX% through XXX and YYY.

XXX Inc., Beijing, China (PR)
Software Engineer Intern March 2010 – June 2010
- Designed and implemented a XXX system to automatically XXX...
- Built a XXX based recommendation system to automatically recommendation ads to customers with XXX and YYY. This system achieves a XXX precision and XXX recall.
- Improved a XXX algorithm to automatically filter noise information from extracted webpages and improved the precision of this algorithm by 50% through replacing XXX module with YYY module.

XXX Inc., Shenzhen, China (PR)
Software Engineer Intern August 2008 – November 2009
- Designed and implemented a XXX system, which consisted of XXX and YYY. Experiments showed that this system was able to XXX.

SKILLS C, C++, Java, Python, .NET, Matlab, Bash, SQL, Linux, Hadoop.

HONORS AND AWARDS
XXX Scholarship, XXX University. 2013
YYY Award, XXX University. 2012

图1-1 一页纸的简历

这里有个小技巧，由于绝大部分科技公司都在加州，特别是北加州硅谷附近，所以如果有朋友在加州的话不妨写朋友的地址。这样做的确可以提高被选中面试的几率，甚至有些小公司可能会跳过一轮电话面试直接邀请你去公司面

试。现在绝大部分面试过程都是通过电话或邮件确定，除了最后给你发offer，招聘的中间部分都不会给你提供地址寄送任何文件。如果公司决定给你offer，你也完全有机会和HR确认/更新你的实际地址。这样做的缺点是，可能去公司面试的时候你需要自己安排机票住宿，如果真到了这一步，权且就当花钱买个机会了。

基本信息之外，对于刚毕业的学生而言最重要的信息包括学校、专业和学位。学习成绩对于大部分公司而言只是加分项，如果GPA低于3.3，可以考虑不包括成绩。当然，如果GPA是你的亮点之一，那也完全可以包含此信息。在这方面，Oracle是出了名的名校控，如果你高GPA、名校出身，基本上半只脚已经踏入了Oracle。相对而言，其他公司并没有这样的要求。此外，对学生而言可以列举一些在校学过的相关课程。只需要列举课程名即可，内容数量以5项左右为宜，课程名尽量选择大家熟知的，比如算法、数据结构、操作系统等，如果是研究生课程，可以加上“高级（advanced）”关键字。

简历中另一个十分关键的内容是之前做过的项目描述，包括工作经历、实习经历或者课程项目经历，这些部分的目的在于展示你具有相关经验，具有较强的技术实力，能够加入团队一起完成一个复杂的项目。这里，我们主要介绍如何选择合适的项目，在描述技巧部分，会进一步介绍如何描述使得你的项目更有趣。就项目内容而言，你需要优先选择和职位描述相关的项目，相关性包括：需要相似的知识，需要使用相同的软件或开发环境，需要类似的编程语言或编程模式，需要实现类似的功能等等。这些项目最能体现你的价值，使得你从一堆申请者中脱颖而出。如果没有相似的经历，那么你可以列举一些比较复杂的项目，突出你的综合能力。这些可以包括：毕业设计、课程的学期作业、网上参与合作的开源项目等等。列举的目的主要是突出你的技术水准优秀，具有与他人合作的能力。就项目种类而言，你需要优先选择业界的工作实习经历，只要它们和你所求职位的描述大体在同一个行业即可。毕竟，有实际的工作经验可以传达下列信息：你对业界有一定的了解，熟悉业界的开发模式和开发周期，能够适应公司的工作环境。

简历的最后部分可以用来列举你的技能，包括熟悉的编程语言、开发环境、技术强项等等。这部分的目的在于让HR能够从简历库中匹配到你的信息。通常，每个职位都有一些技术要求，HR会通过查询关键字，从简历库中选择匹配程度高的简历。这部分可以大大增加简历被匹配上的可能性。列举的原则是，并不需要特别熟悉，只要实际用过就可以在此列举。注意，如果你有幸被安排了面试，一定要回到这个部分，确保你所列举的部分至少都能回答一些基本的问题，千万不要给人做假的感觉。往往，HR并不一定了解你项目部分的描述是否与职位要求一致，因此，你这部分所列举的技能需要尽量用业界标准的语言，列举名词即可。

1.1.3 描述技巧

描述的技巧主要体现在项目描述方面。项目描述主要突出你做了什么，实现了什么样的目的。项目名称一般需要让读者大致了解你做了什么，然后以如

下模版，“通过……开发方式（或者技术），做了……，最终实现了……的结果”，描述项目的具体内容。就英文简历而言，一般以过去时为主，以动词开头，描述你做过了什么，实现了什么目的。举例如下：

Software Engineering Internship，XX Company，6/1/2014 – 9/1/2014

Interned with the server team.

Implemented a distributed access control algorithm in C++, which improved login time by around 50%.

英文简历常见的另一个问题是如何翻译专有名词，比如在国内大学获得的种种荣誉等。此时，一定要参考网络资料，确保读者的理解和你想要表达的意思一致。甚至可以用一句话简单描述这是怎样的荣誉，或者用百分比表示只有top的学生才能获得该项荣誉。举例如下：

Excellent Student Award（top 3%）

Granted to recognize overall outstanding performance.

另一个很好学习途径，就是去LinkedIn上看一些优秀人士的个人主页，特别是你想去的公司的工程师的背景。一般都会很清晰地构建他的目标和过去经历，不需要照搬内容，但可以给你启发，当你去投这些公司时，你就知道什么样的简历更容易被他们欣赏。

1.2 面试

根据面试的不同对象，在招聘过程中你可能需要面对 HR、技术面试官和老板。针对不同的角色，你应该准备不同的面试方式。具体分析如图1-2所示。[1]

[1] 这个表格主要是针对美国求职而言的。中国的程序员并不一定是这种形式，这里只是希望读者可以参考一下北美的面试流程。

面试官	他想要知道	可以向其询问
人力资源	你是谁？你的职业兴趣是什么？	公司的整体信息和组织结构；开放的职位
工程经理	你做过什么项目？具备什么技能？对什么项目感兴趣？	团队的职责是什么？团队目前和将来会做什么项目？团队在找什么样的人？
产品经理	对公司产品的观点/反馈/建议。如果让你设计一款产品，你会怎么做？	公司的下一步产品是什么？公司面临的关键性挑战是什么？公司中的工程师如何与产品经理交互？
午餐面试者	你之前或当前公司/学校的团队怎么样？你喜欢他们吗？为什么？你怎么融入公司的文化？	公司中的团队是怎么样的？他们有哪些团队构建活动？
工程师	如果你能够提出合理的清晰的问题，如果你能够有效地表达你解决问题的规划，如果你最终能够解决这个问题——那么．请展示你的代码	一般的工作流程是怎样的？公司使用的技术栈是什么？
架构师	你是否能够以可扩展的方式解决问题？你是否能够认清系统设计中的关键权衡？	公司使用的技术栈是什么？公司如何使用这些技术来解决现实世界的问题？

图1-2 面试须知

1.2.1 HR

HR是你与公司的连接点。通常，HR负责安排协调面试，主要通过邮件联系。HR也有可能会直接打电话联系你，目的是了解你的基本情况，包括身份、毕业/离职时间等。HR通常会介绍职位要求和公司的基本情况，并且在面试当天接待你，了解你是否有其他面试安排或者其他公司的offer。此外，HR还负责面试你的沟通能力，向老板反馈性格方面与团队的契合度以及对公司感兴趣的程度。因此，每次与HR的沟通也需要热情、职业。适合向HR了解的信息包括：公司的整体氛围、面试的流程安排、最近公司人员流动情况等。

1.2.2 技术面试官

技术面试官主要负责衡量你的技术水平，以及判断你是否符合职位要求。总体而言，对科技公司，技术面试官的意见最为重要。技术面试包括电话面试和现场面试，前者主要偏向概念性的问答，也包括通过协作网站直接写代码等。现场面试通常包括白板写代码，解决一个算法问题或者设计问题等。本书的主要目的就是帮助你通过这轮面试。

一些面试的小技巧如下：一定要先沟通，明确自己了解题意，不要过分考虑或者欠考虑。首先可以给出一个比较容易想到、但并不是最优的解决方案，再逐步优化。在思考的时候也要把思路讲出来，哪怕不是很成熟的方案。一旦遇到困难，可以先自己设法解决，如果五分钟没有思路，可以向面试官求助。适当的提示并不会影响你面试的最终结果。当开始写程序的时候，尽量注意语法格式、变量命名等，避免写伪代码，越接近真实代码越好。写完以后自己检查下有没有明显的错误，可以列举几个简单的测试数据，与面试官一起检验一下整个运行过程。

面试是一个合作解决问题的过程，沟通一定是面试的关键：需要通过沟通展示你的逻辑性、理解能力和表达能力。在面试的最后，通常对方会给你提问的机会，你可以问的问题包括：团队平时使用什么样的技术，通常的工作压力和工作时间，公司最让人兴奋的地方；在当前职位工作了多少年，面试官之前的工作经历与现在相比有什么异同等等。

面试考察的基本功，包括以下方面：

- 程序风格：能正确使用缩进，括号要对齐，变量名可以起的有意义；

- 编码习惯：异常检查，边界处理；
- 沟通：让面试官时刻明白你的意图，不要闭着眼睛不停地写。因为你的算法未必对。对了你也未必写得出来。中间稍微有点问题，你就失败了。对于面试官来说，他根本不知道你的解题进行到哪一步了；
- 测试：主动写出合理的测试用例（Test case），一些常见的用例，如null检查。一般你没写的话，面试官会让你写，但如果你主动写出来，说明你有好的习惯，容易加分。

技术面试的流程通常如下，可供参考。

当你拿到一个具体问题时，可以按照如下流程回答：

1．明确题意：通过与面试官交流明确需要解答的问题。这部分主要为了让自己放松心态，并且给面试官留下你具有良好团队意识和交流能力的印象。

2．描述大体思路：描述你打算用什么算法，什么数据结构。主要是为了让面试官了解你的思维过程，如果你给出的解答与他想要的答案偏差太多，可以及时纠正。同时，描述思路也给了你自己思考的机会。

3．实现算法：先处理边界条件。对于重要的算法模块，加一些注释或者与面试官进行交流。目的是让面试官始终了解你在做什么，算法框架是什么。

4．跑一个测试：用一个测试用例走一遍你写的程序。目的在于和面试官一起确保你的算法是有效的，可以在过程中及时发现并纠正自己的错误。同时，给面试官留下你有写单元测试（unit test）习惯的良好印象。

5．描述算法复杂度，回答面试官的问题。

1.2.3 老板

团队的老板通常最后一个出场面试，或者陪同面试者一起吃午饭。老板负责收集整理所有人的反馈，并且决定是否发offer。通常而言，老板可能不会问过于技术的问题，而是侧重考察你的协作沟通能力。老板的问题可能包括：如何面对工作中的难题/压力，你之前做过的项目，为什么适应这个职位等等。与老板沟通需要表现出你对他们团队的热情，并且在回答中尽量体现自己为什么适合这个职位。面试是一个相互的过程，通过与老板的面试，你需要了解这些问题：团队的成员构成，一般情况下项目如何分配，老板对你的期望，老板对团队在公司中发展的一些展望等等。

另外，准备一些常见的行为问题：比如你有没有过失败的经历，如果你老板给你不喜欢的任务怎么办，你想象中要成为什么样的人。这里一方面可以结合自身精力，另一方面多关注公司的介绍页面，包括公司创始人背景、企业文化、招聘的要求。这些都可以提前做好功课，尽量体现出来你的激情、负责、勤奋等优秀品质。

1.3 Offer

如果走到这一步，那么祝贺你，你成功了！在这一步，你需要一些谈判技巧，为自己争取更多的利益。首先，你要做的是与HR核对信息，包括你的地址、入职时间等。

通常，HR会简单介绍你的待遇福利，当你确认无误后，HR会生成正式文档让你签名。在这个阶段，你可以让HR解释offer条约中你不理解的部分，并且协商你的待遇。协商的最常见方式是，当你有其他公司的offer，你希望最想去的公司能够match其他公司的最高值。注意，在这个阶段，HR是与你站在一条战线上：HR也不希望你轻易地拒绝他们的offer。因此，你完全有理由提出你自己的要求。一般在你有其他公司offer的情况下，HR都能争取到一些更多的利益。从争取难度而言，入职时的签字奖金最容易争取，股票和基准工资则比较难有提升。当然，工资待遇是重要的一方面，但在你决定是否接受offer的时候，综合考虑公司的发展前景、团队在公司中的地位、老板与你交流时你的感受、团队氛围等等也是必不可少的因素。

对于美国的绝大部分公司，offer上都不会写雇佣时间，这意味着双方都可以随时终止合同。通常情况下，大公司不会轻易裁员，哪怕裁员也会有一定的

补助。另一方面，这也说明你可以随时离职，甚至在入职之前，也即毁约。一般来说，不建议这种做法：更合理的做法是尝试与HR沟通，告知对方自己还有其他的面试/offer，需要推迟一段时间做决定。如果实在万不得已，有其他更好的选择，你需要尽早与HR沟通，希望对方理解。一定不要拖到最后告诉对方自己不去了，这样的做法很不职业，也不礼貌。

当你接受offer之后，可以向老板要一些材料，自己先准备一下，以便工作开始的时候能够更快上手。一般新到一个公司都会有数周甚至数月的上手时间，团队会专门有人帮助你了解他们的项目。新的旅程就此开始！但这仅仅是开始，未来也许是更大的挑战，能不能融入到团队，能不能抵抗住压力，工作内容是否符合自己的兴趣，这些都是未知数。所以我们说没有绝对正确的选择，只要用你的才华和汗水付出才有实际意义！

1.4 常见问题

问题1：如何知道一些靠谱的公司？

首先，了解一下“牛人”都选择去哪些公司。如果公司名气不大，可以去流量排名上去看他处于什么地位，公司有没有上市，它的融资规模，还可以从Linkedin看它的员工是否优秀。

在硅谷，大家非常热情地谈创业谈机会，我们也通过自己的一些观察和积累，看到了不少最近几年才涌现出来的热门创业公司。给大家一个列表，这个是华尔街网站的全世界创业公司融资规模评选（http://graphics.wsj.com/billion-dollar-club/）。它本来的标题是“Billion Startup Club”（十亿美金创业公司俱乐部），不到一年的时间，截至到2015年1月17日，现在的排名和规模已经发生了很大的变化，如图1-3所示。

首先，估值在10Billlon（百亿美金）的公司达到了7家，而一年前一家都没有。其次，第一名是中国人家喻户晓的小米，第三，前20名中，绝大多数（8成）在美国，在加州，在硅谷，在旧金山！比如Uber、Airbnb、Dropbox、Pinterest。第四，里面也有不少以相似模式取得成功的公司，比如Flipkart就是印度市场的淘宝，Uber与Airbnb都是共享经济的范畴。所以大家还是可以在移动（Uber）、大数据（Palantir）、消费级互联网、通信（Snapchat）、支付（Square）、O2O App里面寻找大的机会。

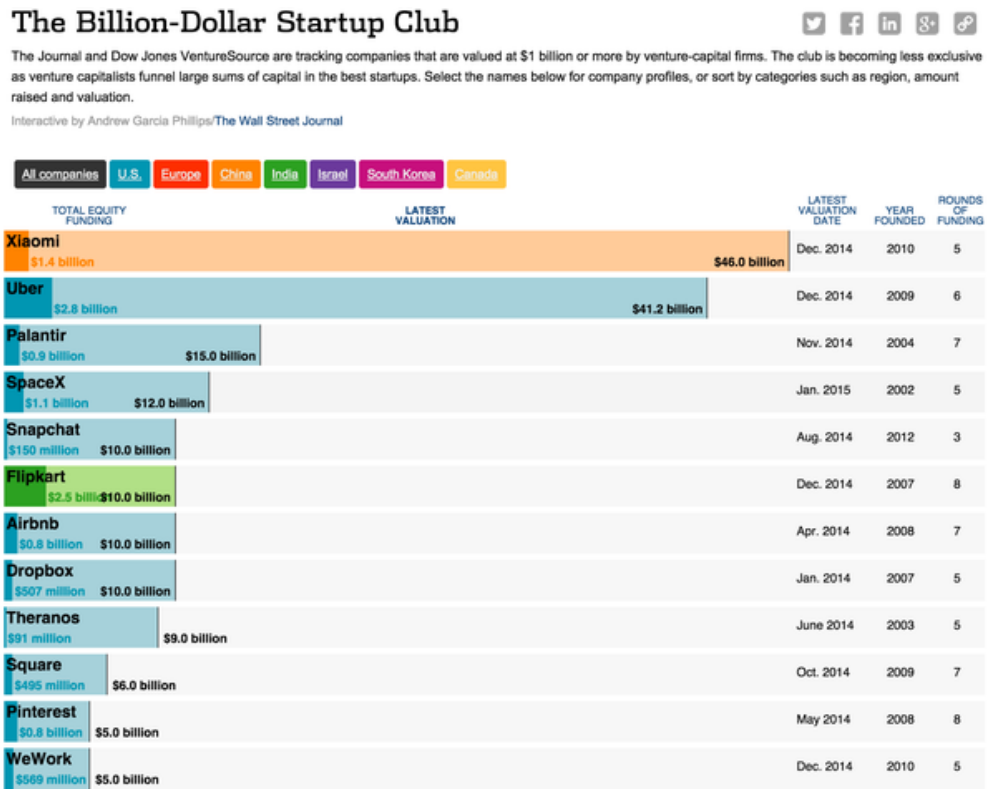

图1-3 十亿美金创业公司俱乐部

问题2：硅谷的Startup有什么技术方案？

分类介绍如下，如图1-4所示。

编程语言：Python、Scala、Swift for iOS、C/C++、Java等主流开发语言；

前端：Play、Video.js、Jade、HTML5等；

框架容器：Docker、Mesos、Vagrant等；

日志监控：Datadog、Sumologic、Akka、Kafka等；

后台数据处理：Hive、Scalding、EMR、Big Query等；

图 1-4 硅谷常用技术矩阵

虚拟机：EC2等云服务；

服务器：Nginx等；

配置工具：NPM、Zookeeper、Puppet、Gradle等；

信号通知：SQS等；

代码托管：Github、SVN等版本控制工具；

数据存储：Cassandra、MySQL、S3、Redshift等；

内部工具：Phabricator、Slack、Google Docs、RelateIQ、Jira等协作工具。

以Coursera为例，作为创业公司，Coursera力图保持敏捷和高效。从技术上来说，所有的内容都是在基于AWS开发，可以想象随意启动云端服务，做一些实验。公司大致分成产品组、架构组和数据分析组。因为公司比较新，所以没有什么历史遗留迁移的问题。大家大胆地使用Scala作为主要编程语言，采用Python作为脚本控制，比如产品组就是提供课程产品，里面大量使用Play Framework、Javascript的Backbone作为控制中枢。而架构组主要是维护底层存储、通用服务、性能和稳定性。笔者所在的数据组，一部分是对商业产品、核心增长指标做监控、挖掘和改进；另一部分是搭建数据仓库，完善与各个部门的无缝数据流动，也用到很多技术；例如使用Scalding编写MapReduce程序，也有人做AB testing框架、推荐系统，尽可能用最少人力做有影响力的事情。其实除了开源世界，Coursera也积极使用第三方的产品，比如Sumologic做日志错误分析，Redshift作为大数据分析平台，Slack做内部通信。而所有的这些的目标就是想解放生产力，把重心放到用户体验、产品开发和迭代上去。

问题3：什么时候才知道准备好面试？

可以从如下几个方面来衡量。

算法是否过关，是否能写出递归和动规；

Coding是否过关，是否能在编译器中写出Bug Free；

Design是否过关，是否能给出Tradeoff；

项目经历整理，能够流利说出架构、难点、自己的贡献；

加分项：Github、Blog、参与Open Source。

问题4：如何拿到美国工作签证？

要来美国工作，一般是要求具有H1B（工作签证）的身份，获得该身份需要有雇主向劳工局提出申请（sponsorship）。根据现在的形式，每年的名额都是一抢而空，那么这就需要抽签决定。在4月1日之前提出申请，4月1日之后开始抽签，如果是在美国获取硕士以上的学位，可以有优先级，抽取概率更高，而根据今年情况，普通的抽中几率是50%。即便没有抽中，如果是拥有美国的硕士学位，可以使用OPT照样工作，并且可以省社会安全税（Social Security Tax）。而如果是海外的学位，只能等来年再抽签。

像Google、Facebook这种全球性公司，他们也会提供其他国家办公室的机会，工作一年后再通过L1或者H1B继续到美国工作。另外如果你抽中了H1B，那么如果以后再跳槽，则可以通过办理转职（transfer）沿用之前的名额，而不需要再次抽签。H1B每3年可以续一次，最多6年。如果H1B到期时已经申请绿卡，那么还是可以延长H1B的有效时间，直至绿卡生效。

问题4：我不是算法大牛，不是ACM队员，听说Google，Facebook有很多牛人才能进，那我怎么能拿到好Offer？

首先是要有信心，算法不是想象那么难。原因有如下两个方面：

第一：因为面试常见的算法就那么几种。只要你努力去总结归类相似题目，只做很少的题，就可以举一反三掌握很多的题。不要盲目关注数字。做题质量非常重要。标准是：你做过的题目，让你再做一次，你就能“完美解决”。

第二：即使你知道一道题的解法，你未必能写好。因为你可能每次写出来的程序都很随性，这样会漏洞百出。程序员是一个非常讲究严谨性的职业，如果你在总结题目的时候能够找到这些题目的模板，把模板提炼好，碰到类似的题目，可以一边写模板，一边想想怎么在模板上做一点简单的改动。这样既节省时间又保证不会出错。

问题5：从其他专业转做计算机专业的该怎么准备？

1．让自己更专业。比如，你的简历只能放和计算机有关的东西，其他东西再牛也不能发挥用处（比如学生会主席）。如果简历还是很空，就多去做项目。实在没项目，就把非计算机专业的项目改写得更接近。

2．在某一方面达到工程师实力。临时转行时间短，找一个容易入手的准备，如果你以前做的事情跟数据有关，就申请数据分析师（Data Analyst）。如果你本来就会一些基本技术，可以做前端，用JavaScript、HTML和CSS，去真正做一个自己博客。如果对产品感兴趣，就玩Django、Ruby on Rails这样的网站框架，了解一个网站是怎么搭建起来的。如果对移动开发感兴趣，就写几个在iOS或者Android上的App。这样做的好处是，简历不空，如果问到，可以驾轻就熟。如果不相关，坦诚相见，说由于我是转行的，在这方面不熟悉，可以尝试一下。也可以直接告诉面试官，我是转行的，我对计算机很感兴趣，做了这么这么几个项目。于是面试官不会问你难题。而实际上你早就准备好了，超出面试官预期，会得到一个面试高分！

问题6：面试时候如何表现自己体现沟通能力？

首先你要站在面试官的角度思考问题。面试官要招你进去当同事，你希望同事怎么样?

你可以反复和面试官交流自己的想法，得到面试官认可以后再动手写。可以讲讲你是怎样想到这个思路的。从而展现你的沟通能力。记住，你并不是要说服他接受你的想法，而是要把你的想法解释给他听。面试官提出质疑的时候，第一，不要觉得面试官什么都不懂，怎么这都不知道（其实他只是看一下你是否真的懂）；第二，面试官比你经验丰富得多，很有可能就是你犯错了，赶紧想想是不是真的有问题。

问题7：面试中出了Bug怎么办？

避免Bug很重要，这个需要我们平时不断地练习，按照上述的方法准备，还是可以避免一些“坑”的。但碰巧你可能不在状态，写出了Bug被面试官指出，是不是就挂了呢?

首先别担心，出Bug很正常，也许面试官来面试你之前正在Debug。衡量一个程序员能力的标准，并不是他能想出多牛的算法，而是程序员在遇到问题的时候分析和解决问题的能力。而出Bug的时候，正是展现你是否是一个合格程序员的时候！

Debug的流程如下所示：

1．通过测试用例定位Bug所在位置；

2．不要立即修改代码，重新梳理逻辑。因为很有可能还有其他Bug；

3．走完所有逻辑之后，心里有数怎么改了，再动手开始改；

4．用测试用例再走一次新的代码；

5．在整个过程中，不停地告诉面试官你在做什么（在不影响正常写程序的情况下）。

这样，成功排解Bug，不但不会减分，还会因为你优秀的Debug能力和与此同时展现出来的沟通能力而加分。

问题8：如何做出最后选择Offer？

在考虑Offer之前，先对公司做个研究，比如这家公司是什么规模，产品是什么，Glassdoor员工如何评价的，你的职位你喜欢吗？这就跟选学校一样，如果选错了，也是需要走很多弯路。我们个人的参考是首先这公司是上升期的，产品是否有爱，团队是否比较强，能否学到东西。对公司分类，例如Hortonworks这种是纯技术性的，面向企业级的，可能没多少人知道，而Uber是大众消费性，很多朋友都用过。现在的热点是移动互联网，大家也可以多考虑这一块。

如果上市的公司，会给限制性股票，分3～4年行使期权，创业公司一般给期权，不同就是限制性股票是白送的，不需要自己掏腰包，期权需要自己买入，不同时期价格不同，但股票交的税非常高，有些期权是长期避税的。最后也要考虑你的兴趣和对风险的承受能力，如果去大公司做个螺丝钉，实现共产主义生活也无可厚非。去小公司压力大，成长快。但也要做好失败的准备，看看当年Zynga教训。

了解了面试的基本过程，让我们回到正题，按章节梳理技术面试的要点。

1.5 工具箱

本小节列出求职、面试和考虑Offer的整个过程中，可以参考的一些有用的网站和资源。

1．求职

Glassdoor
http://www.glassdoor.com

LinkedIn

http://www.linkedin.com/jobs

Indeed
http://www.indeed.com/

CareerBuilder
http://www.careerbuilder.com/

Monster
http://www.monster.com/

2．代码

A Short List of DevOps Tools
http://newrelic.com/devops/toolset

TopCoder
http://www.topcoder.com/

Google CodeJam
https://code.google.com/codejam

CodeChef
http://www.codechef.com/

HackerRank
https://www.hackerrank.com/

ACM ICPC
http://icpc.baylor.edu/

hackathon.io
http://www.hackathon.io

Hacker League
https://www.hackerleague.org

Hackathon Hero
http://www.hackathonhero.com/

3．行业趋势

Y Combinator

https://news.ycombinator.com/jobs

AngelList
https://angel.co/jobs

Reddit
https://www.reddit.com/r/Jobbit

StackOverflow
http://careers.stackoverflow.com/

The Distributed Developer Stack Field Guide
http://sites.oreilly.com/odewahn/dds-field-guide/

4．代码质量

Continuous Integration
http://martinfowler.com/tags/continuous%20integration.html

Java Code Review Checklist
http://java.dzone.com/articles/java-code-review-checklist

Stop More Bugs with our Code Review Checklist
http://blog.fogcreek.com/increase-defect-detection-with-our-code-review-checklist-example/

Embedded System Code Review Checklist
http://users.ece.cmu.edu/~koopman/pubs/code_review_checklist_v1_00.pdf

Typesafe Project & Developer Guidelines
https://gist.github.com/jboner/3864996

List of tools for static code analysis
http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis

Security Source Code Analysis Tools
https://www.owasp.org/index.php/Source_Code_Analysis_Tools

5．团队合作开发

Slack
https://slack.com/

Google Apps (GMail, GChat)
https://www.google.com/work/apps/business/

IRC
http://en.wikipedia.org/wiki/Internet_Relay_Chat

Yammer
https://www.yammer.com/

Asana
https://asana.com/

Microsoft Project
http://products.office.com/en-us/project/project-and-portfolio-management-software

JIRA
https://www.atlassian.com/software/jira/

bugzilla
https://www.bugzilla.org/

GitHub Issues
https://github.com/blog/831-issues-2-0-the-next-generation

Confluence
https://www.atlassian.com/software/confluence

DropBox
https://www.dropbox.com

Google Drive
https://www.google.com/drive/

Zoho
https://www.zoho.com

RelateIQ
https://www.relateiq.com/

BambooHR
http://www.bamboohr.com/

6．薪水

Crunchbase

Wealthfront Startup Salary & Equity Compensation
https://www.wealthfront.com/tools/startup-salary-equity-compensation

AngelList Salary and Equity Data
https://angel.co/salaries

Salary.com
http://www.salary.com/

PayScale Salary Calculator
http://www.payscale.com/salary-calculator

CareerBuilder Salary Calculator
http://salary.careerbuilder.com/

Wolfram Alpha Salary Comparison
http://www.wolframalpha.com/examples/SalariesWages.html

PayScale Cost of Living Calculator
http://www.payscale.com/cost-of-living-calculator

CNN Money Cost of Living Calculator
http://money.cnn.com/calculator/pf/cost-of-living/

An Introduction to Stock & Options for the Tech Entrepreneur or Startup Employee
http://www.scribd.com/doc/55945011/An-Introduction-to-Stock-Options-for-the-Tech-Entrepreneur-or-Startup-Employee

Startup Equity for Employees
http://www.payne.org/index.php/Startup_Equity_For_Employees

An Engineer's Guide to Stock Options
http://blog.alexmaccaw.com/an-engineers-guide-to-stock-options

The Impact of Dilution
https://blog.wealthfront.com/impact-of-dilution/

Term Sheet: Liquidation Preference
http://www.feld.com/archives/2005/01/term-sheet-liquidation-preference.html

I have a job offer at a startup, am I getting a good deal?

http://venturehacks.com/articles/job-offer

The 14 Crucial Questions About Stock Options
https://blog.wealthfront.com/stock-options-14-crucial-questions/

How much are startup options worth?
http://www.danshapiro.com/blog/2010/11/how-much-are-startup-options-worth/

The Equity Equation
http://paulgraham.com/equity.html

How I negotiated my startup compensation
https://keen.io/blog/29904565692/how-i-negotiated-my-startup-compensation