

应届生2018校园招聘 求职大礼包

Google 篇

应届生论坛 Google 版:

<http://bbs.yingjiesheng.com/forum-242-1.html>

应届生求职大礼包 2018 版-其他行业及知名企业资料下载区:

<http://bbs.yingjiesheng.com/forum-436-1.html>

应届生求职招聘论坛 (推荐):

<http://bbs.yingjiesheng.com>

目录

一、Google 公司简介.....	4
1.1 公司概况.....	4
1.2 Google 名字由来.....	4
1.3 Google 价值观.....	5
1.4 Google 技术.....	5
1.5 Google 企业文化.....	6
1.6 Google 公司产品和服务.....	6
1.7 Google 管理团队.....	12
第二章 Google 笔试题目分享.....	12
2.1 Google 谷歌 2014 校园招聘在线笔试题目（全，供参考）.....	12
2.2 Google 2013 校园招聘笔试题心得.....	26
2.3 GOOGLE 中山大学笔试归来.....	30
2.4 2010.09.27-谷歌笔试题详尽版.....	30
2.5 Google 笔试归来.....	33
2.6 google 北大笔试题.....	34
2.7 google 浙大笔试题.....	35
2.8 google 交大笔试题.....	36
2.9 google 笔试集锦.....	38
2.10 google 的 21 道 GLAT 考试.....	42
2.11 Google 惯用“整蛊题”.....	46
2.12 2009 年 google 笔试题选择题部分.....	53
第三章 Google 面试经验分享.....	54
3.1[面试] 软件开发工程师全英语面试.....	54
3.2[面试] google 16 Summer Intern.....	55
3.3[面试] 软件工程师面试.....	55
3.4[面试] 记一次 Google 面试经历.....	55
3.5[面试] 字符串转置匹配 - 面试 Google 工程师必问的 15 道面试题.....	57
3.6[面试] Google 两次面试的经验（phone+onsite）.....	59
3.7 陈广琛：面试体验：Google 篇.....	60
3.8 一次谷歌面试趣事.....	62
3.9 [面试] 软件工程师面试经验.....	64
3.10google 应聘 associate, map quality, 或 SQE.....	64
3.11GoogleTAM 面试记.....	65
3.12 谷歌面试问题.....	66
3.13 谷歌疯狂面试题.....	67
3.14Google 面经小全.....	68
3.15google 面试试题详解.....	68
3.16Google 员工出书大谈公司面试心得.....	70
第四章 Google 求职综合经验.....	71
4.1 我的 Google 求职随感.....	71
4.2 我的找工作之路-google 篇.....	72
附录：更多求职精华资料推荐.....	74

内容声明:

本文由应届生求职网 YingJieSheng.COM (<http://www.yingjiesheng.com>) 收集、整理、编辑, 内容来自于相关企业的官方网站及论坛热心同学贡献, 内容属于我们广大的求职同学, 欢迎大家与同学好友分享, 让更多同学得益, 此为编写这套应届生大礼包 2018 的本义。

祝所有同学都能顺利找到合适的工作!

应届生求职网 YingJieSheng.COM

一、Google 公司简介

Google 官方网站:

<http://www.google.com>

1.1 公司概况

Google 公司 (Google Inc., NASDAQ: GOOG), 是一家美国的上市公司 (公有股份公司), 于 1998 年 9 月 7 日以私有股份公司的形式创立, 以设计并管理一个互联网搜索引擎; Google 公司总部位于加利福尼亚山景城, 在全球各地都设有销售和工程办事处。Google 网站于 1999 年下半年启动; 2004 年 8 月 19 日, Google 公司的股票在纳斯达克 (Nasdaq) 上市, 成为公有股份公司。Google 公司的总部称作 “Googleplex”, 位于美国加州圣克拉拉县的山景城 (Mountain View)。在共创办人拉里·佩奇退下后, Novell 公司的前任行政总裁, 埃里克·施密特 (Eric E. Schmidt) 博士, 成为了 Google 公司的行政总裁。

Google 成立的第一步始于 Google 创始人 Larry Page 和 Sergey Brin 在斯坦福大学的学生宿舍内共同开发了全新的在线搜索引擎, 然后迅速传播给全球的信息搜索者。Google 目前被公认为全球规模最大的搜索引擎, 它提供了简单易用的免费服务, 用户可以在瞬间得到相关的搜索结果。当您访问 www.google.com 或众多 Google 域之一时, 您可以使用多种语言查找信息、查看股价、地图和要闻、查找美国境内所有城市的电话簿名单、搜索数十亿计的图片并详读全球最大的 Usenet 信息存档 - 超过十亿条帖子, 发布日期可以追溯到 1981 年。用户不必特意访问 Google 主页, 也可以访问所有这些信息。

2006 年 4 月 12 日, Google 公司行政总裁埃里克·施密特在北京宣布该公司的全球中文名字为 “谷歌” (有报道指出取义 “丰收之歌”, 不过亦有报道指出取义 “山谷之歌”)。同时, Google 公司于 2006 年 2 月 15 日在台湾地区登记之分公司取名为 “美商科高国际有限公司”。此前, 在一份中国国际经济贸易仲裁委员会域名争议解决中心裁决书中, 公司被称为 “科高公司”。该公司亦拥有 “谷歌.cn”、“谷歌.中国”、“咕果.com” (但不拥有 “咕果.中国” 及 “咕果.公司”) 等中文域名。尽管中文用户在除其英文名外更常称 Google 为 “古狗” 或 “狗狗”, 其中文域名 “古狗.com”、“古狗.cn”、“古狗.中国” 等均已被其他公司抢注。此外, Google 在北京的分公司曾使用 “咕果” 作为合同签订以及网络招聘的中文译名。北京时间(UTC+8)2006 年 4 月 17 日凌晨 1 时左右, Google 简体中文网站正式出现 “谷歌” 字样 (其他地区依旧仅显示 “Google”)。Google 中国对 “谷歌” 的解释是 “播种与期待之歌, 亦是收获与欢愉之歌”, 并称此名称是经 Google 中国的全体员工投票选出。谷歌” 发布不久, 即遭到很多用户的批评。随后, 部分中文用户发起反对 “谷歌” 的网上签名活动。但 Google 中国坚持使用 “谷歌” 作为正式名称, 但在实际中仍然存在 “谷歌”、“Google 中国” 混用的现象。

1.2 Google 名字由来

Google 公司选用 “Google” 一词用来代表在互联网上可以获得的海量的资源。“Google” 一词源于单词 “Googol”, 据说是因拼错而产生的。“Googol” 指的是 10 的 100 次幂, 写出的形式为数字 1 后跟 100 个零。该词现在也可以用作动词, 例如 “google 某物” 的意思是在 google 搜索引擎上搜索 “某物” 这个关键词! Googol 是由美国数学家 Edward Kasner 九岁的侄子 Milton Sirota 发明的, 后来在数学家 Edward Kasner 和 James Newman 的著作 《Mathematics and the Imagination》中被引用。Google 公司采用这个词显示了公司想征服网上无穷无尽资料的雄

心。Google 公司没有采用 Googol 可能是因为版权的问题，而且当他们注册 Google.com 的时候，Googol.com 已经被注册。Google 词义的另一解释：G 意义为手，00 为多个范围，L 意为长，E 意为出，把它们合一起，意义为：我们 GOOGLE 无论在哪里都能为您找出很长的一大堆您想要的。

1.3 Google 价值观

Google 创始人之一 Larry Page 指出：“完美的搜索引擎要做到确解用户之意，切返用户之需”。就搜索技术的现状而言，我们需要通过研究、开发和革新来实现长远的发展。Google 致力于成为这一技术领域的开拓者。尽管 Google 已是全球公认的业界领先的搜索技术公司，但其目标是为所有信息搜寻者提供更高标准的服务。

为了实现这个目标，Google 一直在孜孜不倦地追求技术创新，突破现有技术的限制，随时随地为人们提供快速准确而又简单易用的搜索服务。要完全了解 Google，您需要了解 Google 在重新定义个人、企业和技术人员对互联网的看法的整个过程的方方面面。

1.4 Google 技术

Google 秉持着开发“完美的搜索引擎”的信念，在业界独树一帜。所谓完美的搜索引擎，就如公司创始人之一 Larry Page 所定义的那样，可以“确解用户之意，切返用户之需”。为了实现这一目标，Google 坚持不懈地追求创新，而不受现有模型的限制。因此，Google 开发了自己的服务基础结构和具有突破性的 PageRank™ 技术，使得搜索方式发生了根本性变化。

Google 的开发人员从一开始就意识到：要以最快的速度提供最精确的搜索结果，则需要一种全新的服务器设置。大多数的搜索引擎依靠少量大型服务器，这样，在访问高峰期速度就会减慢，而 Google 却利用相互链接的 PC 来快速查找每个搜索的答案。这一创新技术成功地缩短了响应时间，提高了可扩展性，并降低了成本。这也是其他公司一直在效仿的技术。与此同时，Google 从未停止过对其后端技术的改进，以使其技术效率更高。

Google 搜索技术所依托的软件可以同时进行一系列的运算，且只需片刻即可完成所有运算。而传统的搜索引擎在很大程度上取决于文字在网页上出现的频率。Google 使用 PageRank™ 技术检查整个网络链接结构，并确定哪些网页重要性最高。然后进行超文本匹配分析，以确定哪些网页与正在执行的特定搜索相关。在综合考虑整体重要性以及与特定查询的相关性之后，Google 可以将最相关最可靠的搜索结果放在首位。

PageRank 技术：通过对由超过 50,000 万个变量和 20 亿个词汇组成的方程进行计算，PageRank 能够对网页的重要性做出客观的评价。PageRank 并不计算直接链接的数量，而是将从网页 A 指向网页 B 的链接解释为由网页 A 对网页 B 所投的一票。这样，PageRank 会根据网页 B 所收到的投票数量来评估该页的重要性。

此外，PageRank 还会评估每个投票网页的重要性，因为某些网页的投票被认为具有较高的价值，这样，它所链接的网页就能获得较高的价值。重要网页获得的 PageRank（网页排名）较高，从而显示在搜索结果的顶部。Google 技术使用网上反馈的综合信息来确定某个网页的重要性。搜索结果没有人工干预或操纵，这也是为什么 Google 会成为一个广受用户信赖、不受付费排名影响且公正客观的信息来源。

超文本匹配分析：Google 的搜索引擎同时也分析网页内容。然而，Google 的技术并不采用单纯扫描基于网页的文本（网站发布商可以通过元标记控制这类文本）的方式，而是分析网页的全部内容以及字体、分区及每个文字精确位置等因素。Google 同时还会分析相邻网页的内容，以确保返回与用户查询最相关的结果。

Google 的创新并不限于台式机。为了确保通过便携式设备访问网络的用户能够快速获得精确的搜索结果，Google 还率先推出了业界第一款无线搜索技术，以便将 HTML 即时转换为针对 WAP、i-mode、J-SKY 和 EZWeb 优化的格式。

1.5 Google 企业文化

在 Google，要求工程师们每周都花一天时间在个人感兴趣的项目上。这种近乎强制性的要求造成 Google News 之类的新服务品种出现，根据 Nielsen NetRatings 的数据显示，这项服务现在每个月都能吸引 710 万浏览者，同时也导致了社区网络站点 Orkut 的出现，目前它已经被整合到整个搜索网站之中。Google 里的每一个人都充满了故事：与你共进午餐的人或许发明了你在使用的编程语言；坐在你隔壁的同事或许为你的研究生课程编写过教材；和你一起打台球的那个人或许开发过你的桌面浏览器。

Google 内部依然保持了当年.COM 时代的奢华待遇。公司提供员工免费餐点，早中晚餐全包。若要往来于办公室之间，员工可骑乘 Segway 电动滑板车，或者 GreenMachine 车——一种适合于 11 岁儿童的玩具车。

巧克力、懒人球(一种开会用的座椅，球状)以及巨型积木随处可见，使这里更像是托儿所。公司里面设有牙医与家庭医生，请育婴假的员工可照领 75% 的薪水，婴儿出生后两周内，公司每天补贴 50 美元当作员工的“坐月子”津贴。

除此之外，公司还提供免费的班车和渡轮服务接载雇员上班，这些交通工具都有无线互联网服务，方便员工在上下班时也可以工作。

1.6 Google 公司产品和服务

Google 搜索引擎

公司产品 Google 是全世界最受欢迎的搜索引擎，使用一种自创的称为 PageRank™ (网页级别) 技术来索引网页，索引是由程序“Googlebot”执行的，它会定期地请求访问已知的网页新拷贝。页面更新愈快，Googlebot 访问的也愈多。再通过在这些已知网页上的链接来发现新页面，并加入到数据库。索引数据库和网页缓存大小是以兆兆字节 (terabyte) 来衡量的。A culture has grown around the very popular search engine and the word to google has come to mean, "to search for something on Google."

由于 Google 已经成为最流行的搜索引擎之一，很多网站管理员十分热衷于跟踪他们网站的排名，并试图解释他们排名变化的原因。因此，现在已有不少网站提供服务，意图在一些高流量的讨论区内刻意加入商业网站的链接，从而使该网站在 Google 的排名提高。这种“发明”虽然的确有一定成效，但这种收取客户金钱，在第三者的讨论区上大卖广告，一方面对讨论区的读者做成困扰，也侵害了讨论区的商业利益；而这种做法也明显违反了商业道德。Google 已将大量先前的测试服务整合为搜索功能的一部分(如 Google 计算器)。在 Google 中搜索 what is the answer to life, the universe and everything? (什么是生命，宇宙以及所有一切事物的答案?) 将会得到智能化的搜索结果 42。(这是著名科幻小说《银河系漫游指南》的情节，被释义为是人工智能达到一定高度的表现，即机器能释读人类的语言。)

Google Web API

Google Web API(网络应用程序接口或网络服务)是 Google 为注册的开发者提供的公共接口.使用 Simple Object Access Protocol(SOAP,简单对象访问协议),程序员可以依据 Google 搜索结果开发搜索服务和进行数据挖掘.同样的,网虫们也可以访问页面缓存然后对页面提出建议.

缺省的,一个开发者每天只能有 1,000 次搜索请求.这个程序仍然处于测试中.Google 是很少的几个把其结果通过公共网络应用程序接口公开给大众的搜索引擎;Technorati 是另外一个这样做的公司. Google 这项服务的一些流行应用包括, Google Alert 最新资料快报 FindForward,它同时也是一个调查 Google 跳舞情况的工具,它监视著 Google 蜘蛛在万维网上的活动情况.

Google Book Search

2004 年 8 月, Google 开始提供一项名为 Google Print 的新服务, 现已更名为 Google Book Search. 该工具可以在搜索页面提供由内容出版商提供的书本内容的搜索结果, 并提供连向购买书本的网页以及内容相关广告. Google 会限制可查阅书本的页数. [7], 不过有人已经发现破解方法. 至 2005 年 12 月, 该服务仍然处于 Beta 阶

段。这个服务与 A9.com 提供的很类似。

2004 年 12 月, Google 扩展了 Google Print 的功能。[8] 其书本包括了一些著名大学和一些公共图书馆, 包括密歇根大学, 哈佛大学的 Widener 图书馆, 斯坦福大学的格林图书馆), 牛津大学的牛津大学图书馆以及纽约公共图书馆。根据这些大学图书馆和图书的出版状况, Google 计划十年内将有约 1500 万本位于公共领域的书上线。

2005 年 11 月 17 日, Google 将此服务更名为 Google Book Search。

Gmail

2004 年 4 月 1 日愚人节, Google 宣布推出 1GB 空间的电子邮件服务 Gmail, 次年 4 月 1 日更是增加为 2GB 以上, 并增加了多语言入口以及富文本编辑功能。这个邮箱会利用广告词服务扫描邮件内容以显示特定的广告来营利。无论如何, 最初发布的消息引发了业界的怀疑。Google 的产品副主管 Jonathan Rosenberg 稍后向英国广播公司的新闻重申“我们对 Gmail 的发布是认真的。”

Gmail 邮箱拥有 5000MB 以上的存储空间。比起大部分其他免费的网络邮箱服务都要大很多——例如, 微软的 Hotmail 和雅虎邮箱。为了回击 Gmail, 这两家公司稍后推出了邮箱升级计划, Hotmail 英文免费邮箱将由 2MB 升级到 5GB, 雅虎免费邮箱由 4-6MB 升级到 100MB, 后来雅虎免费邮箱更升级到 1GB, 到了后期雅虎甚至将信箱容量调整为无上限。其他的邮箱服务提供商也紧跟其后, 有一些邮箱现在已经比 Gmail 还要大了。

2004 年 8 月 25 日, 这项服务开始进行内部测试, 公众不能自由注册。不管怎么样, 一些 Blogger 用户还有收到已有的 Gmail 用户邀请的人已经可以使用他了。由于在类似 swap 的交换系统上 Gmail 邀请被人用钱和物品交换, Google 后来改变了他的政策: 不允许用钱来换 Gmail 邀请。2007 年 2 月 14 日, Google 结束了 Gmail 测试期, 正式向公众开放注册。

有许多人批评 Gmail 的隐私策略/ 其中有很多是批评“残留的电子邮件拷贝可能会在我们的系统上保留一段时间, 甚至是你把它从邮箱里删除或是终止你的帐号后也是这样。”这一条款的。很多人相信这意味着 Google 会有意的永久存档已被删除的邮件副本。Google 稍后宣称他们将“使用时将尽快地使合理有效的从我们的系统里删除信息”。然而, 很多的批评仍然反对 Google 计划在邮箱内添加自动扫描邮件的文本关联广告。

Blogger

Blogger 是全球最大、最多人使用的博客系统。2003 年, Google 接管了 Pyra 实验室及其 Blogger 服务。Google 使得先前需要收费的一些功能对用户免费。Blogger 工具及服务, 使得发布 weblog 变得更加简单。用户不需要书写任何代码或是安装任何软件和脚本。而且, 用户可以自由的改变 blog 的设计方案。

Orkut

Orkut 没有在 Google 的页面上被提及。Orkut 是一个服务提供商, 它由 Google 工程师们创建并维护。Orkut 是一种社会性网络服务, 在 Orkut 用户可以留下他们的个人或专业信息, 创建与朋友之间的关系或者因为共有的兴趣爱好加入虚拟社团。

Google Notebook

Google Notebook 是 Google 提供的一项服务, 让使用者方便地储存及整理从网路上收集的资料, 并且利用其共享功能让使用者将自己的笔记公开给其他人浏览。中文版于 2007 年 4 月正式推出。

Picasa

2004 年 7 月 13 日, Google 接管了 Picasa 公司软件的开发, Picasa 软件可以管理共享数字图像。Picasa 同时被整合进 Google 的 Blogger 内。现在它是免费的, 而且提供对中文的全面支持。

Chrome 谷歌浏览器

Google Chrome, 中文名为谷歌浏览器, 是一个由 Google 公司开发的开放源代码网页浏览器。本软件的代码是基于其他开放源代码软件所撰写, 包括 WebKit 和 Mozilla, 目标是提升稳定性、速度和安全性, 并创造出简单且有效率的使用者界面。软件的名称是来自于又称作“Chrome”的网络浏览器图形使用者界面 (GUI)。

软件的 beta 测试版本在 2008 年 9 月 2 日释出, 提供 43 种语言版本, 目前仅适用于 Microsoft Windows 的 XP 及 Vista 平台, 并不支持 Windows 2000 或更早期的版本。Mac OS X 和 Linux 版本正在研发中, 并将于日后推出。

Google 桌面搜索

2004 年 Google 推出本地资源搜索工具, 需要安装一个不到 2M 的程序在硬盘, 便可通过桌面搜索搜索本地硬

盘里的文档。支持的搜索内容包括：

Outlook Email Outlook Express Word Excel Powerpoint Internet Explorer AOL Instant Messenger Netscape Mail/Thunderbird 企业版还包括 Lotus Notes Email Netscape/Firefox/Mozilla PDF Music Images Video QQ MSN Messenger，通过使用插件，Google Desktop Search 还可以支持任何一种文件，前提是，你必须有这种文件类型的插件（无论 Google 开发的、或是你自行开发的、或是第三方软件）。Google 桌面搜索推出后，Yahoo!、MSN 以及一些中国大陆搜索网站也相继推出了桌面搜索工具。

之前据报 Google 正在开发一款下载工具，内部开发名称为 Puffin，它可以搜索本地文件。Puffin 似乎是为了迎战下一代微软视窗操作系统 Windows Vista 中将提供的文件和网络搜寻能力而开发的——Windows Vista 中的这项功能将与 Google 的技术核心万维网搜索业务产生直接的竞争。

2004 年下半年谷歌公司推出了这一桌面搜索工具，它的正式名称是 Google Desktop Search，这个工具主要用来通过关键字方式搜索安装了该软件的计算机硬盘上的 MS Office 和 TXT 文档。

2005 年 3 月 8 日该软件英文正式版推出正式版，同时推出中文、韩文测试版。该英文正式版产品增加了众多功能，包括插件的使用。5 月 26 日该软件推出中文正式版，支持 QQ 和 MSN Messenger 聊天记录搜索。Google Blog。该软件对中文的支持程度较微软和雅虎的桌面搜索工具更好且软件体积更小，速度也较其它软件快。

Google 工具栏

Google 工具栏是一个免费的 IE 插件。功能包括：在不打开 Google 网页的情况下随时搜索并查看相关页面信息；查看 Google 对网页的 PageRank；阻止自动弹出窗口；自动填写表单；用不同颜色标识关键字。有人指称使用它会带来安全风险，因为它会在无用户干预的情况下自动更新。

英语版的 Google 工具栏可以参与 Google Compute 计划，这是一个由美国斯坦福大学进行的帮助研究蛋白质折叠、误解、聚合及由此引起的相关疾病的分布式计算工程（参见 Folding@home）。他浏览器，如 Mozilla Firefox，Opera 和 Safari，有提供相同功能的内建搜寻工具。Mozilla Firefox 还有一个专门的 Google 工具栏：Googlebar，它是被独立开发的，并不被 Google 或 Mozilla Firefox 开发者所支持。它在官方的 Google 工具栏上做了扩展，惟一没被取代的功能是 Google PageRank™。Google 功能也已经被苹果电脑的新操作系统 OS X 的 Safari 所内建。

2005 年 7 月 7 日，Google 推出官方 Firefox 版本工具栏的测试版，同时也有多种语言版本可以选择。该 Firefox 版本工具条提供了 IE 工具栏 3Beta 版包括的拼写检查、字词翻译和“自动链接”等新功能。

Google 工具栏目前已经推出 3.0 版本，其中主要的新功能包括了拼写检查、字词翻译和“自动链接”。自动连接功能能在鼠标指向的相关超链接下显示相关信息，如鼠标指向一个 ISBN 号，则在下方就可以显示出 Amazon 网上书店中这本书的相关信息。目前，该连接功能只适用于美国。最近，也有人批评该项创新功能是破坏了网页创建者的网页自主权。因为此项功能实际是通过修改原网页 HTML 代码实现。

Google Web Accelerator

2005 年 5 月 4 日，谷歌公司推出一款新的软件：Google Web Accelerator。它是一款为宽频连接设计的软体。该软体可以利用 Google 位于世界各地的服务器而加速网页的访问，其原理类似于破网软件，透过架设本地代理，把浏览器的连接请求通过此代理访问实现。与一般的破网软件不同的是，Google Web Accelerator 会向 Google 网站传送回诸如页面请求、临时 cookies 之类的数据，通过与位于 Google 服务器上的版本比对只下载更新的部分，而且因为一般访问 Google 网站较其他网站快，这也加快了网络访问。该软件更可以对经常访问的网页进行存档以加速访问。Google 网页加速器可以利用 Google 全球的计算机网络的处理能力，提高网站的加载速度。还能够向计算机发送前对网站数据进行压缩。Google 的这一系统存储用户经常访问的网站的拷贝，并自动获取这些网站的新数据，当用户请求访问这些网站时，浏览器仅仅需要加载网站上更新的部分。对于隐私问题，该软件不会访问任何 https 网站，也可以自定义某些网站不允许加速。Google 宣称不会搜集或转售个人信息。

该软件目前默认支持 Mozilla Firefox 和 Internet Explorer 浏览器。对于其他浏览器需要手动配置代理服务器。2005 年 5 月 8 日，谷歌公司在中文网页上取消了该软件的下载。Google 官方宣称原因是已经达到其服务器最大负载。但有人推测可能另有原因，例如因它类似破网软件的性质而不受中国政府欢迎。无论如何，Google 虽然去除了下载连接，但软件没有删除。用户仍然可以通过或直接下载该软件。目前，该软件在中国大陆的部分地区不能起到加速作用，估计是受到了网络封锁。

Google Mars

3月13日, Google公司发布了称为Google Mars的网上服务, 纪念天文学家罗威尔(Percival Lowell)的诞辰(1855年3月13日~1916年11月16日)。

Google Maps

Google Maps提供各种地图服务, 包括局部详细的卫星照片。2005年6月20日, Google Maps的覆盖范围从原先的美国、英国及加拿大扩大为全球。在2006年底更加入香港街道。因某一不合理条款的限制, 中国版google本地搜索提供地图的是一家国内的公司: mapabc而不是Google的数据。

Google Moon

2005年7月20日, 谷歌公司发布了称为Google Moon的网上服务, 纪念阿波罗11号于1969年7月20日登月36周年。此服务以之前发布的Google Maps作基础, 卫星数据则来自NASA。用户可使用Google Moon观看月球凹凸不平的表面地形, 当把地图放到最大时, 月球表面会变成一片乾酪。

Google SketchUp

Google于2006年3月14日宣布收购3D绘图软体SketchUp及其开发公司@Last Software。SketchUp是一套以简单易用著称的3D绘图软体, Google收购SketchUp是为了增强Google Earth的功能, 让使用者可以利用SketchUp建造3D模型并放入Google Earth中, 使得Google Earth所呈现的地图更具立体感、更接近真实世界。使用者更可以透过一个名叫Google 3D Warehouse的网站寻找与分享各式各样利用SketchUp建造3D模型。

Google维护著一个重要的新闻组存档, 它被叫做Google网上论坛(即从前一个叫做DejaNews的独立网站)和一个图像搜索服务(被叫做“Google图像”)。前者保存了几十年内几乎所有的新闻组帖子, 后者的搜索则是以与图片相关的网页的文本, 图片的标题为基础进行的, 图片被以合理使用原则缓存进了Google服务器。

Google 新闻

Google有一个测试版的自动化新闻服务, 2004年9月“Google新闻”包括有美国版、英国版、德国版、法国版、西班牙版、意大利版、新西兰版、印度版、澳洲版、韩国版、日本版、中国大陆版、中国台湾版和中国香港版。为了公正客观没有偏见地报道任何新闻, Google新闻的产生是完全由电脑算法决定的, 没有人类编辑参与其中。在2006年1月, 中国版的“Google新闻”改名为“Google资讯”, 内容都没有经过审查。后来, Google推出适合中国使用的Google.cn资讯版, 但其中的内容就经过了审查。即使这样, 也常在打开约30秒后访问中断或无法显示图片。

该服务包括在过去30天内所含语言新闻网站上出现新闻的存档, 不同的国家有不同数量的新闻来源; 对于英语它包括大约4,500个新闻源, 其他语言比较少一些。并且提供新闻的大约头200个字和一个指向全文的连接。一些需要先订阅才能阅读的网站; Google新闻标题中还会有明显的提示信息。

Google新闻提供搜索服务, 结果可以以新闻发生日期(这样就不会再对新闻发生的时间感到困扰了)或相关性成类排序(也可以直接分类查看)。在英语版中, 有一个可以选择对应国家的选项。

还可以使用关键字订阅Google新闻快讯。这样, 当与关键字相关的新闻发生时, Google新闻会发出一封电子邮件通知订阅者。

2005年3月10日Google新闻增加了自定义功能, 用户可以自己随意定义自己喜欢看的新闻, 甚至不同语言的新闻也可以混和在一页内。这是网络新闻提供方式的一个重大革新。

Google 网页目录

Google网页目录是一个包括了世界多种语言网页的目录集。在网页目录里面的网页内容一般不会被翻译为其他语言, 而总是包括其语言在万维网中的内容的。网页目录功能与网页搜索是集成的, 当搜索网页时, 相关网页在目录中的内容会以链接的形式在搜索结果中显现。点击链接就可以找到在同一个目录下相似网页或其它类似分类, 这当你不确定到底要找什么时是非常有用的。当搜索范围涵括太广, 使用网页目录可缩小搜索于指定范围。例如察看“中文/新闻/杂志”分类子目录, 则可知道有哪些中文杂志有网页。网页目录可略去类似但无关的网页。如检索“大学”, 将搜索范围设定“教学机构”分类, 即可略去像“大学书城”、古书里“大学”、论语的内容。网页目录只包括经编辑群审核过网站。因为网页目录是在开放式目录(Open Directory)工程下运作的。网页重要性排列是网页级别技术及人工的结合。Google还可辨出常用重要网站, 排放在目录前面(用粗体字标出)提升网页

搜索效率并借由绿线长短表明网页评级。(参见 PageRank™)

Google Answers

2002年4月,Google启动了名为"Google Answers"的新服务.Google Answers是传统搜索功能的扩展—用户不用自己搜索内容,他们请专家搜索然后付费.顾客问问题,并为问题提供一个相应的价钱,然后研究者们回答他们的问题.研究者们经过程序的筛选以测试他们的水平和交际能力.问题的价格从\$2到\$200不等;Google从中提取25%回扣,剩下的归研究者所有,他们还要付\$0.50的列出费.一旦一个问题被回答了,它的答案对所有人就都可以免费浏览了.这个服务在2003年5月开始公共测试,但到2006年由于受到雅虎知识的影响,决定停止这项服务。

Google Talk

Google的即时聊天软件,于2005年8月24日发布.原先用户必须先拥有一个Gmail帐号才能登录并使用这个软件,但现在已经向所有用户开放.原本只有英文版,现在繁体及简体中文已经推出,功能还很简单.具有VoIP语音通话功能,界面简单清新,安装文件不到900K.软件的设计思想是力求简洁,所以尚不具备文件传输和表情图片等功能(但在对话中发布“:)”等文字会被Gtalk软件以彩色显示,可见设计者想到了表情符号的功能).2006年3月,Google Talk的文字聊天记录可以被保存在用户的Gmail账户里,并且Gmail用户可以在网页上使用Google Talk的部分聊天功能。

Google Local

2005年9月5日,Google公司在中国推出本地搜索服务,连地址也本地化,Google本地搜索:查找本地公司与服务.英文版是Google Local,到目前为止,Google Local已经在美国、英国、日本和加拿大开始运行,中国是GOOGLE开启这项服务的第五个国家,中文版是Google本地.为中国版google本地搜索提供地图的是一家国内的公司:mapabc

Google Scholar

2004年11月,Google发布"Google Scholar",这是一个学术文献资源搜索引擎.搜索结果根据"相关性"排列,这与Google网站使用的PageRank非常类似.2006年1月11日,Google公司宣布将Google学术搜索扩展至中文学术文献领域。

Google Special

这个搜索服务提供了包括美国政府,Linux,BSD,麦金塔和微软四个特别领域的搜索服务

Friend Connect

可以让网站管理员在他们的网站增加一些社会性的功能,全部工作只需要简单复制粘贴几段代码即可,不需要复杂的编程技术。

网站设计者需要做的仅仅是从Friend Connect的功能列表中进行选择,这些功能包括注册、邀请、用户相册、信息发布、评论,以及对OpenSocial第三方应用的支持。

Friend Connect能够使互联网任何位置的用户实现简单的互联,并使‘任何应用、任何网站、任何好友’的理念成为现实。

Google Video

2005年1月25日,Google公司推出Google Video服务,该服务可以通过Google网站搜索网络上的各种视频文件,或最近播出的电视节目.Google Video根据关键词提供相关的视频内容下载或播放连接,并提供视频内容的预览画面.目前该服务仍处于Beta测试阶段.在电视节目搜索方面,目前仅能搜索美国播出的电视节目.Google公司还推出了一款专用的视频文件上传软件,允许使用者将本机的视频文件上传到Google Video中.自从收购Youtube后,公司将Google Video资源整合到了前者。

YouTube

YouTube成立时间为2005年2月,创办人为现任行政总裁的29岁台湾华人陈士骏(Steve Chen)和现为技术执行长的27岁Jawed Karim.总部地点位于美国加州,提供视频短片共享服务,网站口号为“Broadcast Yourself”(广播您自己),是web2.0中的播客,其网站在短时间内迅速蹿红,成为当时Alexa全球排名第10位的网站.而且它的访问率在同类站点里占有60%以上的市场,超过了微软公司的MSN Video30%,Google Video10%.YouTube一般禁止用户上传有侵权与色情内容的短片.YouTube凭一位香港中学生拍摄的“巴士阿叔”短片被台湾、香港

传媒广泛报道后迅速在大中华地区走红。而该片的点击率更高达 1,965,539 次（截至 2008 年 5 月 18 日）。

Google 曾公开承认 Google Video 错失良机，无法和 Youtube 竞争。2006 年 10 月，Google 公司以 16.5 亿美元宣布收购 YouTube，这是迄今为止 Google 公司花费最大的一笔收购，不过也有分析人士称 Google 公司不得不面对今后网友短片发布知识产权保护相关的纠纷。

igoogle

Google 提供了创建个性化 iGoogle 网页的功能，从而可以让您快速、迅捷地浏览 Google 和网络上的重要信息。您可在此自定义网页上选择和组织内容，例如：

- 最新的 Gmail 邮件
- Google 资讯和其它顶级新闻来源的头条新闻
- 天气预报、股价和电影放映时间表
- 可从任何计算机快速访问自己喜爱网站的书签
- 您自己添加的栏目：可包含您从整个网络上找到的内容。

Google 音乐搜索服务

2009 年 3 月 30 日，Google 正式提供对中国地区的音乐搜索服务支持，Google 免费提供各种高品质正版音乐的搜索视听和下载，结束了互联网版权混乱的局面。

Google 谷歌拼音输入法

谷歌拼音输入法是由 Google 中国实验室开发，于 2007 年 4 月 2 日发布，10 月 25 日成为 Google 的正式产品之一。

谷歌搜索服务器虚拟版 GSAve

谷歌搜索服务器(Google Search Appliance, GSA)能够对企业、高校等拥有的网页、文档、数据库等相关信息进行高效的通用搜索，支持的文件格式达 220 余种。随着 GSA 最新的 5.2 版本的发布，谷歌同时发布了一个运行在虚拟机中的纯软件版的 GSA——谷歌搜索服务器虚拟版 (GSAve)。GSAve 可以使用连接器管理工具，抓取和索引到非 Web 形式的网络资源；使用 Feed API 和元数据搜索功能，您可以为自己的网络论坛 (BBS) 建立搜索功能；安全搜索功能支持多种身份验证方式，使用户在搜索结果中只见到自己有权访问的文档；同时专门为小规模文档集设计了专门的网页排序算法；用户可以定制 GSA 的搜索结果界面，甚至以 XML 格式的形式整合到应用中去。

Google Earth

Google 地球 (Google Earth) 是一款 Google 公司开发的虚拟地球仪软件，它把卫星照片、航空照相和 GIS 布置在一个地球的三维模型上。

Google Earth 于 2005 年向全球推出，被“PC 世界杂志”评为 2005 年全球 100 种最佳新产品之一。用户可以通过一个下载到自己电脑上的客户端软件，免费浏览世界各地的高清晰度卫星图片。

Google Earth 使用了公共领域的图片、受许可的航空照相图片、KeyHole 间谍卫星的图片和很多其他卫星所拍摄的城镇照片。甚至连 Google Maps 没有提供的图片都有。分为免费版与专业版两种。

2009 年 2 月，Google Earth 推出最新版 5.0，新增加了全球各地的历史影像、探索海洋和音视频录制的简化浏览，三大功能让人惊艳。另外，部分以前阴影覆盖的敏感区域也得到解禁。

Google Street View

Google 在 Where 2.0 大会上宣布了 Google Maps 的"Street View" (街道视图) 功能。"StreetView"功能允许用户在 GoogleMaps 上面浏览街道的实景图片，用户就像置身于实地。虽然当前这个功能只支持美国的旧金山、迈阿密、丹佛、拉斯维加斯及纽约五个主要城市，但毫无疑问，就像 Google Maps 的图像更新一样，未来这个功能肯定会支持更多的地区。

Google 进入中国的早期历史

迅速便捷的 Google 搜索技术，自其诞生不久就受到了中国网民的关注。因 Google 早期主要通过作为 Yahoo.com 首页的搜索引擎向公众提供服务，Yahoo.com 极速高效的搜索效果迅即引起了敏锐的中国人的关注。Google 诞生后不久的 1999 年 12 月，中国即出现了首个“剥离 Yahoo 的繁杂分类信息”，只保留一个简单 Yahoo

搜索框的网站：world2shop.com/search。虽然技术含量不高，但仍可谓首家使用 Google PageRank 技术的中文搜索网，Google 在中国的首个站点。此外，Google 优异的技术和搜索效果在中国网民中吸引了一大批热爱 Google 技术文化的粉丝，他们亲切地称呼 Google 为“谷歌”“狗狗”“狗哥”，以各种活动，建立各种相关网站来表达他们对全球领先搜索技术的尊敬。

1.7 Google 管理团队

高层管理团队

- * Eric Schmidt 博士，董事长兼首席执行官
- * Larry Page，创始人兼产品总裁
- * Sergey Brin，创始人兼技术总裁
- * Shona Brown，业务运营高级副总裁
- * W. M. Coughran, Jr.，工程事务副总裁
- * David C. Drummond，公司发展事务高级副总裁
- * Alan Eustace，工程与研究高级副总裁
- * Urs Hölzle，运营高级副总裁，Google Fellow
- * Jeff Huber，工程事务副总裁
- * Omid Kordestani，全球销售及业务拓展高级副总裁
- * George Reyes，高级副总裁兼首席财务官
- * Jonathan Rosenberg，产品管理事务高级副总裁
- * Elliot Schrage，全球通联及公共事务副总裁

Google 管理团队

- * Tim Armstrong，广告销售事务副总裁
- * Nikesh Arora，欧洲运营副总裁
- * Sukhinder Singh Cassidy，亚太和拉丁美洲地区运营副总裁
- * Vinton G. Cerf，副总裁兼首席互联网顾问
- * Salar Kamangar，产品管理副总裁
- * Marissa Mayer，搜索产品与用户体验副总裁
- * Norio Murakami，Google 日本副总裁兼总经理
- * Susan Wojcicki，产品管理副总裁

如果你想了解更多 Google 的概况，你可以访问 Google 官方网站：<http://www.google.com>

第二章 Google 笔试题目分享

Google 的笔试题基本上都是和程序有关的。另外还有一些比较 BT 的逻辑思维题。

2.1 Google 谷歌 2014 校园招聘在线笔试题目（全，供参考）

本文原发于应届生 BBS，发布时间：2014 年 8 月 4 日

地址：<http://bbs.yingjiesheng.com/thread-1871839-1-1.html>

【Google 谷歌大中华区 2014 校园招聘在线笔试经历了三轮，分别在 2013 年 9 月 14 日（历时）、9 月 23 日、10 月 12 日举行，google 原链接：<https://code.google.com/codejam/contests.html>】

Google of Greater China Test for New Grads of 2014

Practice Round China New Grad Test 2014 【 2013 年 9 月 14 日】

Problem A. Bad Horse

ProblemAs the leader of the Evil League of Evil, Bad Horse has a lot of problems to deal with. Most recently, there have been far too many arguments and far too much backstabbing in the League, so much so that Bad Horse has decided to split the league into two departments in order to separate troublesome members. Being the Thoroughbred of Sin, Bad Horse isn't about to spend his valuable time figuring out how to split the League members by himself. That what he's got you -- his loyal henchman -- for.

InputThe first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with a positive integer M on a line by itself -- the number of troublesome pairs of League members. The next M lines each contain a pair of names, separated by a single space.

OutputFor each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is either "Yes" or "No", depending on whether the League members mentioned in the input can be split into two groups with neither of the groups containing a troublesome pair.

Limits $1 \leq T \leq 100$.

Each member name will consist of only letters and the underscore character.

Names are case-sensitive.

No pair will appear more than once in the same test case.

Each pair will contain two distinct League members.

Small dataset $1 \leq M \leq 10$.

Large dataset $1 \leq M \leq 100$.

Sample

Input

```
2
1
Dead_Bowie Fake_Thomas_Jefferson
3
Dead_Bowie Fake_Thomas_Jefferson
Fake_Thomas_Jefferson Fury_Leika
Fury_Leika Dead_Bowie
```

Problem B. Captain Hammer

ProblemThe Hamjet is a true marvel of aircraft engineering. It is a jet airplane with a single engine so powerful that it burns all of its fuel instantly during takeoff. The Hamjet doesn't have any wings because who needs them when the

fuselage is made of a special Wonderflonium isotope that makes it impervious to harm.

Piloting the Hamjet is a not a job for your typical, meek-bodied superhero. That's why the Hamjet belongs to Captain Hammer, who is himself impervious to harm. The G-forces that the pilot endures when taking a trip in the Hamjet are legen-dary.

The Hamjet takes off at an angle of θ degrees up and a speed of V meters per second. V is a fixed value that is determined by the awesome power of the Hamjet engine and the capacity of its fuel tank. The destination is D meters away. Your job is to program the Hamjet's computer to calculate θ given V and D .

Fortunately, the Hamjet's Wondeflonium hull is impervious to air friction. Even more fortunately, the Hamjet doesn't fly too far or too high, so you can assume that the Earth is flat, and that the acceleration due to gravity is a constant 9.8 m/s^2 down.

InputThe first line of the input gives the number of test cases, T . T lines follow. Each line will contain two positive integers -- V and D .

OutputFor each test case, output one line containing "Case #x: θ ", where x is the case number (starting from 1) and θ is in degrees up from the the horizontal. If there are several possible answers, output the smallest positive one.

An answer will be considered correct if it is within 10^{-6} of the exact answer, in absolute or relative error. See the [FAQ](#) for an explanation of what that means, and what formats of floating-point numbers we accept.

Limits $1 \leq T \leq 4500$;

$1 \leq V \leq 300$;

$1 \leq D \leq 10000$;

It is guaranteed that each test case will be solvable.

Sample

Input	Output
398 98098 490299 1234	Case #1: 45.0000000Case #2: 15.0000000Case #3: 3.8870928

Problem C. Moist

ProblemMoist has a hobby -- collecting figure skating trading cards. His card collection has been growing, and it is now too large to keep in one disorganized pile. Moist needs to sort the cards in alphabetical order, so that he can find the cards that he wants on short notice whenever it is necessary.

The problem is -- Moist can't actually pick up the cards because they keep sliding out his hands, and the sweat causes permanent damage. Some of the cards are rather expensive, mind you. To facilitate the sorting, Moist has convinced Dr. Horrible to build him a sorting robot. However, in his rather horrible style, Dr. Horrible has decided to make the sorting robot charge Moist a fee of \$1 whenever it has to move a trading card during the sorting process.

Moist has figured out that the robot's sorting mechanism is very primitive. It scans the deck of cards from top to bottom. Whenever it finds a card that is lexicographically smaller than the previous card, it moves that card to its correct place in the stack above. This operation costs \$1, and the robot resumes scanning down towards the bottom of the deck, moving cards one by one until the entire deck is sorted in lexicographical order from top to bottom.

As wet luck would have it, Moist is almost broke, but keeping his trading cards in order is the only remaining joy in his miserable life. He needs to know how much it would cost him to use the robot to sort his deck of cards.

InputThe first line of the input gives the number of test cases, T . T test cases follow. Each one starts with a line

containing a single integer, **N**. The next **N** lines each contain the name of a figure skater, in order from the top of the deck to the bottom.

Output For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the number of dollars it would cost Moist to use the robot to sort his deck of trading cards.

Limits $1 \leq T \leq 100$.

Each name will consist of only letters and the space character.

Each name will contain at most 100 characters.

No name will start or end with a space.

No name will appear more than once in the same test case.

Lexicographically, the space character comes first, then come the upper case letters, then the lower case letters.

Small dataset $1 \leq N \leq 10$.

Large dataset $1 \leq N \leq 100$.

Sample

Input

```
2
2
Oksana Baiul
Michelle Kwan
3
Elvis Stojko
Evgeni Plushenko
Kristi Yamaguchi
```

Round A China New Grad Test 2014 【2013年9月23日】

Problem A. Read Phone Number

Problem Do you know how to read the phone numbers in English? Now let me tell you.

For example, In China, the phone numbers are 11 digits, like: 15012233444. Someone divides the numbers into 3-4-4 format, i.e. 150 1223 3444. While someone divides the numbers into 3-3-5 format, i.e. 150 122 33444. Different formats lead to different ways to read these numbers:

150 1223 3444 reads one five zero one double two three three triple four.

150 122 33444 reads one five zero one double two double three triple four.

Here comes the problem:

Given a list of phone numbers and the dividing formats, output the right ways to read these numbers.

Rules:

Single numbers just read them separately.

2 successive numbers use double.

3 successive numbers use triple.

4 successive numbers use quadruple.

5 successive numbers use quintuple.

6 successive numbers use sextuple.

7 successive numbers use septuple.

8 successive numbers use octuple.
 9 successive numbers use nonuple.
 10 successive numbers use decuple.

More than 10 successive numbers read them all separately.

InputThe first line of the input gives the number of test cases, **T**. **T** lines|test cases follow. Each line contains a phone number **N** and the dividing format **F**, one or more positive integers separated by dashes (-), without leading zeros and whose sum always equals the number of digits in the phone number.

OutputFor each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the reading sentence in English whose words are separated by a space.

Limits $1 \leq T \leq 100$.

Small dataset $1 \leq \text{length of } N \leq 10$.

Large dataset $1 \leq \text{length of } N \leq 100$.

Sample

Input

315012233444 3-4-415012233444 3-3-512223 2-3

Output

Case #1: one five zero one double two three three triple fourCase #2: one five zero one double two double three triple fourCase #3: o

Problem B. Rational Number Tree

ProblemConsider an infinite complete binary tree where the root node is 1/1 and left and right childs of node p/q are p/(p+q) and (p+q)/q, respectively. This tree looks like:



is known that every positive rational number appears exactly once in this tree. A level-order traversal of the tree results in the following array:1/1, 1/2, 2/1, 1/3, 3/2, 2/3, 3/1, ...

Please solve the following two questions:

1. Find the **n**-th element of the array, where **n** starts from 1. For example, for the input 2, the correct output is 1/2.
2. Given **p/q**, find its position in the array. As an example, the input 1/2 results in the output 2.

InputThe first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of one line. The line contains a problem id (1 or 2) and one or two additional integers:

1. If the problem id is 1, then only one integer **n** is given, and you are expected to find the **n**-th element of the array.
2. If the problem id is 2, then two integers **p** and **q** are given, and you are expected to find the position of **p/q** in the array.

OutputFor each test case:

1. If the problem id is 1, then output one line containing "Case #x: p q", where x is the case number (starting from 1), and p, q are numerator and denominator of the asked array element, respectively.
2. If the problem id is 2, then output one line containing "Case #x: n", where x is the case number (starting from 1), and n is the position of the given number.

Limits $1 \leq T \leq 100$; **p** and **q** are relatively prime.

Small dataset $1 \leq n, p, q \leq 216-1$; **p/q** is an element in a tree with level number ≤ 16 .

Large dataset $1 \leq n, p, q \leq 264-1$; **p/q** is an element in a tree with level number ≤ 64 .

Sample

Input	Output
41 22 1 21 52 3 2	Case #1: 1 2Case #2: 2Case #3: 3 2Case #4: 5

Problem C. Sorting

ProblemAlex and Bob are brothers and they both enjoy reading very much. They have widely different tastes on books so they keep their own books separately. However, their father thinks it is good to promote exchanges if they can put their books together. Thus he has bought an one-row bookshelf for them today and put all his sons' books on it in random order. He labeled each position of the bookshelf the owner of the corresponding book ('Alex' or 'Bob'). Unfortunately, Alex and Bob went outside and didn't know what their father did. When they were back, they came to realize the problem: they usually arranged their books in their own orders, but the books seem to be in a great mess on the bookshelf now. They have to sort them right now!!

Each book has its own *worth*, which is represented by an integer. Books with odd values of worth belong to Alex and the books with even values of worth belong to Bob. Alex has a habit of sorting his books from the left to the right in an increasing order of worths, while Bob prefers to sort his books from the left to the right in a decreasing order of worths. At the same time, they do not want to change the positions of the labels, so that after they have finished sorting the books according their rules, each book's owner's name should match with the label in its position.

Here comes the problem. A sequence of **N** values **s0, s1, ..., sN-1** is given, which indicates the worths of the books from the left to the right on the bookshelf currently. Please help the brothers to find out the sequence of worths after sorting such that it satisfies the above description.

InputThe first line of input contains a single integer **T**, the number of test cases. Each test case starts with a line containing an integer **N**, the number of books on the bookshelf. The next line contains **N** integers separated by spaces, representing **s0, s1, ..., sN-1**, which are the worths of the books.

OutputFor each test case, output one line containing "Case #X: ", followed by **t0, t1, ..., tN-1** in order, and separated by spaces. **X** is the test case number (starting from 1) and **t0, t1, ..., tN-1** forms the resulting sequence of worths of

the books from the left to the right.

Limits $1 \leq T \leq 30$.

Small dataset $1 \leq N \leq 100$

$-100 \leq s_i \leq 100$

Large dataset $1 \leq N \leq 1000$

$-1000 \leq s_i \leq 1000$

Sample

Input	Output
255 2 4 3 17-5 -12 87 2 88 20 11	Case #1: 1 4 2 3 5 Case #2: -5 88 11

Problem D. Cross the maze

Problem Edison, a robot, does not have a right hand or eyes. As a brave robot, **he always puts his left hand on the wall no matter he walks or turns around**. Because he thinks it is too dangerous, Edison does not walk backward. Assume that Edison has found himself in a square-shaped maze of $N \times N$ square cells which is surrounded by walls from the outside. In the maze, some of the cells are also walls. Edison can only move between two empty cells in four directions, north, south, west and east. In order to get out of the maze, he drafts a plan. He uses his left hand to lean on the wall and goes by following the wall.

Here is the question, is Edison able to get out of the maze in at most 10,000 steps? If he can make it, output the path. By getting out of the maze, he only needs to be in the exit cell. If the starting cell is the same as the exit, Edison won't need to move and can directly get out of the maze.

Input The first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with an integer N . N is the size of the maze. The following N lines, each line contains N characters which may be '.' or '#'. '.' is an empty cell, '#' is a wall. Followed by a line which contains four integers: s_x, s_y, e_x, e_y . (s_x, s_y) means that Edison is standing on row s_x and column s_y as his starting cell, (e_x, e_y) is the exit of the maze. (s_x, s_y) is guaranteed to be at one of the 4 corners of the maze, and Edison can only touch the wall on 4 adjacent cells(not 8) initially. (e_x, e_y) can be anywhere in the maze. Note that the top-left corner is at position (1,1).

Output For each test case, output a line containing "Case #x: y", where x is the case number (starting from 1) and y is "Edison ran out of energy." (without the quotes) if Edison can't reach the exit of the maze in at most 10,000 steps, otherwise y should be the number of steps followed by another line which contains y characters to describe the path (each character should be E for east, S for south, W for west or N for north). There is no character to represent the turning around. We don't care about the turning around steps, please only output the path of how Edison will cross the maze.

Limits $1 \leq T \leq 30$.

$1 \leq s_x, s_y, e_x, e_y \leq N$.

The starting cell and the exit of the maze will always be an empty cell. And the starting cell and the exit of the maze won't be the same.

Small dataset $2 \leq N \leq 10$.

Large dataset $2 \leq N \leq 100$.

Sample

Input	Output
32.##.1 1 2 25.##.#.....#.###....#.1 1 5 33....#....1 1 3 3	Case #1: Edison ran out of energy.Case #2: 22S

Note:

In the 2nd test case after moving 1 cell down from his starting cell, Edison will still be able to lean on the wall at the cell (1,2) by his left hand.

In the third test case, due to Edison can't touch the wall at cell (2,2) initially, so he has to go east in his first step.

Problem E. Spaceship Defence

ProblemThe enemy has invaded your spaceship, and only superior tactics will allow you to defend it! To travel around your spaceship, your soldiers will use two devices: *teleporters* and *turbolifts*.

Teleporters allow your soldiers to move instantly between rooms. Every room contains a teleporter, and rooms are color-coded: if a soldier is in a room with some color, she can use the teleporter in that room to immediately move to any other room with the same color.

Turbolifts allow your soldiers to move between rooms more slowly. A turbolift is like an elevator that moves in many directions. Each turbolift moves from one room to one other room, and it takes a certain amount of time to travel.

Notes about turbolifts:

- Turbolifts are not two-way: if a turbolift moves soldiers from room a to room b, the same turbolift cannot move soldiers from room b to room a, although there might be another turbolift that does that.
- More than one soldier can use the same turbolift, and they do not interfere with each other in any way.

You will be given the locations and destinations of several soldiers. For each soldier, output the minimum amount of time it could take that soldier to travel from his location to his destination.

InputThe first line of the input gives the number of test cases, **T**. **T** test cases follow.

For every test case:

The first line of every test case contains an integer **N**, which is the number of rooms in your spaceship. The rooms are numbered from 1 to **N**. The following **N** lines each contain a string telling the color of the rooms, from room 1 to room **N**. The strings only contain characters a-z (the lower-case English letters) and 0-9 (the number 0 to 9), and the length of each string will be less than or equal to 2.

The next line in the test case is an integer **M**, which indicates the number of turbolifts in your spaceship. The following **M** lines each contain 3 space-separated integers **ai**, **bi**, **ti**, telling us that there is a turbolift that can transport soldiers from room **ai** to room **bi** in **ti** seconds.

The next line in the test case contains an integer **S**, which is the number of soldiers at your command. The following **S** lines each contain two integers: the location and destination of one soldier, **pj** and **qj**.

OutputFor each test case, output one line containing only the string "Case #x:", where x is the number of the test case (starting from 1). On the next **S** lines, output a single integer: on line **j**, the smallest number of seconds it could take for a soldier to travel from **pj** to **qj**. If there is no path from **pj** to **qj**, the integer you output should be -1.

Limits $1 \leq S \leq 100$.

$1 \leq ai, bi \leq N$.

$0 \leq t_i \leq 1000.$ $1 \leq p_j, q_j \leq N.$ **Small dataset** $1 \leq T \leq 10.$ $1 \leq N \leq 1000.$ $0 \leq M \leq 3000.$ **Large dataset** $T = 1.$ $1 \leq N \leq 80000.$ $0 \leq M \leq 3000.$

Sample

Input

33glt3t331 2 2173 2 5671 1 2122 12 34cabl8z031 22 31 18reb7yegr0l0lyeb774 1 192 4 212 5 3174 5 344 7 34 8 2658 6 7134 32 6

Round B China New Grad Test 2014 【2014 年 10 月 12 日】

Problem A. Sudoku Checker

Problem **Sudoku** is a popular single player game. The objective is to fill a 9x9 matrix with digits so that each column, each row, and all 9 non-overlapping 3x3 sub-matrices contain all of the digits from 1 through 9. Each 9x9 matrix is partially completed at the start of game play and typically has a unique solution.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Given a completed $N2 \times N2$ Sudoku matrix, your task is to determine whether it is a *valid* solution. A *valid* solution must satisfy the following criteria:

- Each row contains each number from 1 to $N2$, once each.
- Each column contains each number from 1 to $N2$, once each.
- Divide the $N2 \times N2$ matrix into $N2$ non-overlapping $N \times N$ sub-matrices. Each sub-matrix contains each number from 1 to $N2$, once each.

You don't need to worry about the uniqueness of the problem. Just check if the given matrix is a valid solution.

InputThe first line of the input gives the number of test cases, T . T test cases follow. Each test case starts with an integer N . The next $N2$ lines describe a completed Sudoku solution, with each line contains exactly $N2$ integers. All input integers are positive and less than 1000.

OutputFor each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is "Yes" (quotes for clarity only) if it is a valid solution, or "No" (quotes for clarity only) if it is invalid. Note that the judge is case-sensitive, so answers of "yes" and "no" will not be accepted.

Limits $1 \leq T \leq 100$.

Small dataset $N = 3$.

Large dataset $3 \leq N \leq 6$.

Sample

Input

```
335 3 4 6 7 8 9 1 26 7 2 1 9 5 3 4 81 9 8 3 4 2 5 6 7 8 5 9 7 6 1 4 2 3 4 2 6 8 5 3 7 9 17 1 3 9 2 4 8 5 6 9 6 1 5 3 7 2 8 4 2 8 7 4 1 9 6 3 3
5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 3 5 3 4 6 7 8 9 1 2 6 7 2 1 9 5 3 4 8 1 9 8 3 4
4 2 8 7 4 1 9 6 3 5 3 4 5 2 8 6 1 7 9
```

Problem B. Meet and party

Problem Little Sin lives in a Manhattan-grid city, a 2D plane where people can only go north, west, south or east along the grid. The distance from (x_1, y_1) to (x_2, y_2) is $|x_1 - x_2| + |y_1 - y_2|$.

Little Sin really likes to party and is hoping to host a house party in Manhattan this Sunday. Little Sin has collected a list of people who will attend, and now needs to decide at whose home she will host the party.

Little Sin invited all of the people in several rectangular areas, and all of those people have said yes. A rectangular area is denoted as (x_1, y_1, x_2, y_2) , where $x_1 \leq x_2, y_1 \leq y_2$. People who live in a rectangular area fill all integral points inside it. So there are a total of $(x_2 - x_1 + 1) * (y_2 - y_1 + 1)$ people in the rectangular area (x_1, y_1, x_2, y_2) .

Little Sin knows the coordinates of those rectangular areas. She wants the party to be hosted at the home of one of the people who is attending, but she also doesn't want everyone else to have to travel very far: she wants to minimize the sum of all distances from all attendees' houses to the party. Can you help her?

Input The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case starts with a line containing a single integer: the number of rectangular areas, **B**. **B** lines follow. Each line contains 4 integers: x_1, y_1, x_2, y_2 , denoting the coordinates of a rectangular area of people Little Sin has invited to her party.

Output For each test case, output one line containing "Case #: $x y d$ ", where t is the case number (starting from 1) and (x, y) is the coordinates of the person whose home the party should be hosted. If there are multiple positions with the same minimum total distance, choose the one with the smallest x . If there are still multiple positions, choose the one with the smallest y . The value d is the sum of the distances from all attendees' houses to the point (x, y) .

Limits $1 \leq T \leq 10$.

$|x_1|, |y_1|, |x_2|, |y_2| \leq 109$.

$x_1 \leq x_2, y_1 \leq y_2$.

The rectangular areas within a test case don't intersect.

Small dataset $1 \leq B \leq 100$.

$1 \leq$ Total number of people in each test case ≤ 1000 .

Large dataset $1 \leq B \leq 1000$.

$1 \leq$ Total number of people in each test case ≤ 1000000 .

Sample

Input

```
210 0 2 23 -1 2 -1 20 0 0 0 1 3 1 3
```

Output

```
Case #1: 1 1 1
```

Problem C. Hex

ProblemThis problem was inspired by a board game called Hex, designed independently by Piet Hein and John Nash. It has a similar idea, but does not assume you have played Hex.

This game is played on an **NxN** board, where each cell is a hexagon. There are two players: Red side (using red stones) and Blue side (using blue stones). The board starts empty, and the two players take turns placing a stone of their color on a single cell within the overall playing board. Each player can place their stone on any cell not occupied by another stone of any color. There is no requirement that a stone must be placed beside another stone of the same color. The player to start first is determined randomly (with equal probability among the two players).

The upper side and lower sides of the board are marked as red, and the other two sides are marked as blue. The goal of the game is to form a connected path of one player's stones connecting the two sides of the board that have that player's color. The first player to achieve this wins. Note that the four corners are considered connected to both colors.

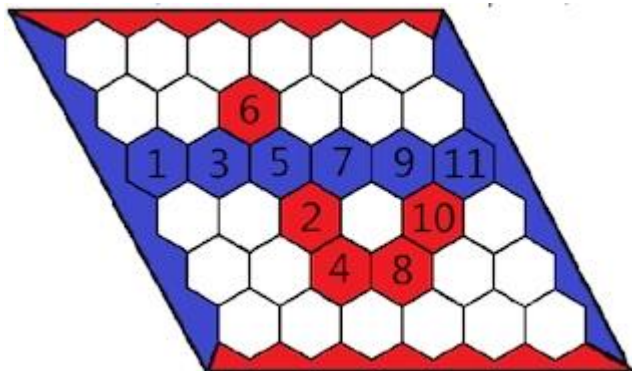
The game ends immediately when one player wins.

Given a game state, help someone new to the game determine the status of a game board. Say one of the following:

- **"Impossible"**: If it was impossible for two players to follow the rules and to have arrived at that game state.
- **"Red wins"**: If the player playing the red stones has won.
- **"Blue wins"**: If the player playing the blue stones has won.
- **"Nobody wins"**: If nobody has yet won the game. Note that a game of Hex can't end without a winner!

Note that in any impossible state, the only correct answer is "Impossible", even if red or blue has formed a connected path of stones linking the opposing sides of the board marked by his or her colors.

Here's a an example game on a 6x6 gameboard where blue won. Blue was the first player to move, and placed a blue stone at cell marked as 1. Then Red placed at cell 2, then blue at cell 3, etc. After the 11th stone is placed, blue wins.



InputThe first line of input gives the number of test cases, **T**. **T** test cases follow. Each test case start with the size of the side of the board, **N**. This is followed by a board of **N** rows and **N** columns consisting of only 'B', 'R' and '.' characters. 'B' indicates a cell occupied by blue stone, 'R' indicates a cell occupied by red stone, and '.' indicates an

empty cell.

Output For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the status of the game board. It can be "Impossible", "Blue wins", "Red wins" or "Nobody wins" (excluding the quotes). Note that the judge is case-sensitive, so answers of "impossible", "blue wins", "red wins" and "nobody wins" will be judged incorrect.

Limits $1 \leq T \leq 100$.

Small dataset $1 \leq N \leq 10$.

Large dataset $1 \leq N \leq 100$.

Sample

Input	Output
71.1B1R2BRBB4BBBBBBB.RRR.RRRR4BBBBBBBRRR.RRRR6.....R...BBBBBB..R.R...RR.....	Case #1: Nobody wins Case #7: Blue wins

Problem D. Dragon Maze

Problem You are the prince of Dragon Kingdom and your kingdom is in danger of running out of power. You must find power to save your kingdom and its people. An old legend states that power comes from a place known as Dragon Maze. Dragon Maze appears randomly out of nowhere without notice and suddenly disappears without warning. You know where Dragon Maze is now, so it is important you retrieve some power before it disappears.

Dragon Maze is a rectangular maze, an $N \times M$ grid of cells. The top left corner cell of the maze is (0,0) and the bottom right corner is ($N-1$, $M-1$). Each cell making up the maze can be either a dangerous place which you never escape after entering, or a safe place that contains a certain amount of power. The power in a safe cell is automatically gathered once you enter that cell, and can only be gathered once. Starting from a cell, you can walk up/down/left/right to adjacent cells with a single step.

Now you know where the entrance and exit cells are, that they are different, and that they are both safe cells. In order to get out of Dragon Maze before it disappears, you must walk from the entrance cell to the exit cell taking as few steps as possible. If there are multiple choices for the path you could take, you must choose the one on which you collect as much power as possible in order to save your kingdom.

Input The first line of the input gives the number of test cases, T . T test cases follow.

Each test case starts with a line containing two integers N and M , which give the size of Dragon Maze as described above. The second line of each test case contains four integers enx , eny , exx , exy , describing the position of entrance cell (enx , eny) and exit cell (exx , exy). Then N lines follow and each line has M numbers, separated by spaces, describing the $N \times M$ cells of Dragon Maze from top to bottom. Each number for a cell is either -1, which indicates a cell is dangerous, or a positive integer, which indicates a safe cell containing a certain amount of power.

Output For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1). If it's possible for you to walk from the entrance to the exit, y should be the maximum total amount of power you can collect by taking the fewest steps possible. If you cannot walk from the entrance to the exit, y should be the string "Mission Impossible." (quotes for clarity). Please note that the judge requires an exact match, so any other output like "mission impossible." or "Mission Impossible" (which is missing the trailing period) will be judged incorrect.

Limits The amount of power contained in each cell will not exceed 10,000.

$1 \leq T \leq 30$.

$0 \leq \text{enx}, \text{exx} < \text{N}$.

$0 \leq \text{eny}, \text{exy} < \text{M}$.

Small dataset $1 \leq \text{N}, \text{M} \leq 10$.

Large dataset $1 \leq \text{N}, \text{M} \leq 100$.

Sample

Input

22 30 2 1 02 -1 53 -1 64 40 2 3 2-1 1 1 21 1 1 12 -1 -1 11 1 1 1

Problem E. Ignore all my comments

Problem Good programmers write fabulous comments. Igor is a programmer and he likes the old C-style comments in `/* ... */` blocks. For him, it would be ideal if he could use this style as a uniform comment format for all programming languages or even documents, for example Python, Haskell or HTML/XML documents.

Making this happen doesn't seem too difficult to Igor. What he will need is a comment pre-processor that removes all the comment blocks in `/*`, followed by comment text, and by another `*/`. Then the processed text can be handed over to the compiler/document renderer to which it belongs—whatever it is.

Igor's pre-processor isn't quite that simple, though. Here are some cool things it does:

The comments the pre-processor reads can be nested the same way brackets are nested in most programming languages. It's possible to have comments inside comments. For example, the following code block has an outer level of comments that should be removed by the comment pre-processor. The block contains two inner comments.
`printf("Hello /* a comment /* a comment inside comment */ inside /* another comment inside comment */ string */ world");`
 After the pre-process step, it becomes:
`printf("Hello world");`
 Igor recognizes comments can appear anywhere in the text, including inside a string `/*...*/`, a constant number `12/*...*/34` or even in a character escape `\/*...*/`
 Or more formally:

text: text-piece text-piece

remaining-text text-piece: char-sequence-without-/* empty-string remaining-text: comment-block

text comment-block: /* comment-content */ comment-content: comment-piece comment-piece

remaining-comment comment-piece: char-sequence-without-/*-or-*/ empty-string remaining-comment: comment-block comment-content char: letters digits punctuations whitespaces

Our pre-processor, given a **text**, removes all **comment-block** instances as specified.

Notes Igor only needs to remove the comment in one pass. He doesn't remove additional comment blocks created as a result of the removal of any comment block. For example: `/*no recursion*/ file header */` should generate: `file header */`
 The `*` character in any `/*` or `*/` cannot be re-used in another `/*` or `*/`. For example the following does **NOT** form a proper comment block: `/*Input`
Input A text document with comment blocks in `/*` and `*/`. The input file is valid. It follows the specification of **text** in the problem statement. The input file always terminates with a newline symbol.

Output We only have one test case for this problem. First we need to output the following line.

Case #1: Then, print the document with all comments removed, in the way specified in the problem statements. Don't remove any spaces or empty lines outside comments.

Limits The input program contains only:

Letters: a-z, A-Z, Digits: 0-9 Punctuation: ~ ! @ # % ^ & * () - + = : ; ' < > , . ? | / \ { } [] _
 Whitespace characters: space,

newline

Small datasetThe small input contains a program of less than 2k bytes.

Large datasetThe large input contains a program of less than 100k bytes.

Sample

Input

```
/*no recursion*/ file header***** Sample input program *****/int spawn_wor
many more. Commented until I figure out why. for (int i = 0; i < worker_count; ++i) { if(!fork()) { /* This is the worker.
main() { printf("Hello /*a comment inside string*/ world"); int worker_count = 0/*octal number*/144; if (spawn_workers(worker
```

Output

```
Case #1:/* file header*****/int spawn_workers(int worker_count) { return 0; }int main() {
{ exit(-1); } return 0;}
```

2.2 Google 2013 校园招聘笔试题心得

本文原发于应届生 BBS，发布时间：2013 年 6 月 18 日

地址：<http://bbs.yingjiesheng.com/thread-1713658-1-1.html>

昨天晚上去蹭了一下 Google 的招聘笔试。其实是去打酱油的，主要是为了感受一下 Google 的出题风格和考试氛围，可以对将来找工作提供些参考。

回来之后本来想回忆一下题目的，结果发现 braveheart89 大大已经贴出了所有的题而且连选项都一字不差，记忆力真心佩服.....以下就根据他写的题目稍微修正一下[1]，然后随便说说好了。（说的也不一定对，欢迎更正。）考试是第一页需要填写个人信息，包括实习经历、获奖情况、工作地点意向（国内、国外还是两者皆可之类，反正对我无用啦-.-）然后就是一个半小时的答题，全部手写。

1、单项选择题

1.1 如果把传输速率定义为单位时间内传送的信息量（以字节计算）多少。关于以下几种典型的数据传输速率：

- 1.使用 USB2.0 闪存盘，往 USB 闪存盘上拷贝文件的数据传输速率
- 2.使用 100M 以太网，在局域网内拷贝大文件时网络上的数据传输速率
- 3.使用一辆卡车拉 1000 块单块 1TB 装满数据的硬盘，以 100km/h 的速度从上海到天津（100km）一趟所等价的数据传输带宽
- 4.使用电脑播放 MP3，电脑的 PCI 总线到声卡的数据传输速率

在通常情况下，关于这几个传输速率的排序正确的是：

- A.4<1<2<3 B.1<4<2<3 C.4<1<3<2 D.1<4<3<2

1.2 对以下程序，正确的输出结果是

```
#define SUB(x,y) x-y#define ACCESS_BEFORE(element,offset,value) *SUB(&element, offset) =valueint main(){ int array[10]= {1,2,3,4,5,6,7,8,9,10}; int i; ACCESS_BEFORE(array[5], 4, 6); printf("array:"); for (i=0; i<10; ++i){ printf("%d", array); } printf("\n"); return (0);}A.array: 1 6 3 4 5 6 7 8 9 10
```

B.array: 6 2 3 4 5 6 7 8 9 10

C.程序可以正确编译连接，但是运行时会崩溃

D.程序语法错误，编译不成功

1.3 在区间 $[-2, 2]$ 里任取两个实数，它们的和 >1 的概率是：

A.3/8 B.3/16 C.9/32 D.9/64

1.4 小组赛，每个小组有 5 支队伍，互相之间打单循环赛，胜一场 3 分，平一场 1 分，输一场不得分，小组前三名出线。平分抽签。问一个队最少拿几分就有理论上的出线希望：

A.1 B.2 C.3 D.4

1.5 用二进制来编码字符串“abcdabaa”，需要能够根据编码，解码回原来的字符串，最少需要多长的二进制字符串？

A.12 B.14 C.18 D.24

1.6 10 个相同的糖果，分给三个人，每个人至少要得一个。有多少种不同分法

A.33 B.34 C.35 D.36

1.7 下列程序段，循环体执行次数是：

```
y=2
```

```
while(y<=8)
```

```
y=y+y;
```

A.2 B.16 C.4 D.3

1.8 下面哪种机制可以用来进行进程间通信？

A.Socket B.PIPE C.SHARED MEMORY D.以上皆可

1.9 下列关于编程优化的说法正确的是：

A.使用编译器的优化选项(如-O3)后程序性能一定会获得提高

B.循环展开得越多越彻底，程序的性能越好

C.寄存器分配能够解决程序中的数据依赖问题

D.现代主流 C/C++编译器可以对简单的小函数进行自动 inline

1.10 一下程序是用来计算两个非负数之间的最大公约数：

```
long long gcd(long long x, long long y) { if( y==0) return 0; else return gcd (y, x%y);}我们假设 x,y 中最大的那个数的长度为 n，基本运算时间复杂度为  $O(1)$ ，那么该程序的时间复杂度为：
```

A. $O(1)$ B. $O(\log n)$ C. $O(n)$ D. $O(n^2)$

2、程序设计与算法

(2.1-2.2 为编程题，需给出代码实现；2.3 为算法设计题，只需设计思路 and 关键步骤伪代码)

2.1 写函数，输出前 N 个素数。不需要考虑整数溢出问题，也不需要使⤵用大数处理算法。

2.2 长度为 n 的数组乱序存放着 0 至 n-1。现在只能进行 0 与其他数的 swap，请设计并实现排序。

2.3 给定一个原串和目标串，能对源串进行如下操作：

- 1.在给定位置插入一个字符
- 2.替换任意字符
- 3.删除任意字符

要求写一个程序，返回最少的操作数，使得源串进行这些操作后等于目标串。源串和目标串长度都小于 2000。

——
以下是我根据各种来源总结的参考答案：

1.1 A USB 2.0 的理论传输极限是 480Mbps[2]，但是按照这个速率就没有选项可选了-.-，所以猜测应该认为是普通 U 盘写数据的 6MB/s，即 48Mbps；

100M 以太网的速率就是 100Mbps；

卡车拉硬盘， $1000 \times 1000 \times 8 / 3600 = 2222$ Mbps，这个应该是最快的；

MP3 在 256kbps 码率下也平均只有 1 分钟 2MB，所以不会超过 0.3Mbps，所以一定是最慢的。

1.2 D 这道题大家走出考场后争议非常大。咱啥也不说，直接进 mingw 跑一下 gcc：



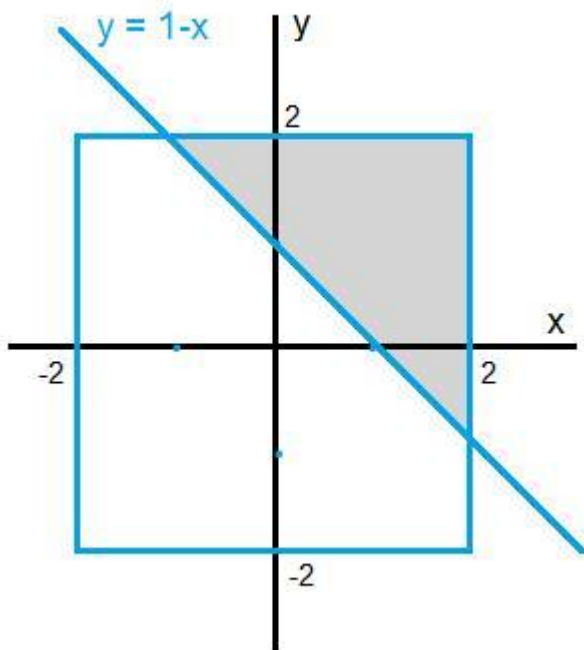
```
MINGW32:~/google1.2
Fishwing@Fishwing-LabPC ~
$ cd google1.2/
Fishwing@Fishwing-LabPC ~/google1.2
$ gcc -o out google1.2.c
google1.2.c: In function 'main':
google1.2.c:9:2: error: lvalue required as left operand of assignment
google1.2.c:10:2: warning: incompatible implicit declaration of built-in function 'printf' [enabled by default]
Fishwing@Fishwing-LabPC ~/google1.2
$
```

gcc 提示的错误是“赋值号的左边操作数需要一个左值”。其原因是调用宏的那句被预处理器替换成了：

```
*&array[5]-4 =6;
```

由于减号比赋值优先级高，因此先处理减号；由于减号返回一个数而不是合法的左值，所以编译报错。

1.3 C 这道题我是蒙对的-.- 标准做法是先画出 $y=1-x$ 的线，上侧阴影部分就是 $y>1-x$ ，其所占比例为 9/32：



1.4 B 这道题我从 A 开始凑胜负表，直到 B 凑出结果就 OK 了。

1.5 B 这道题需要对 abcd 进行 Huffman 编码。首先根据权值建立 Huffman 树，得到最优编码：

a=0, b=10, c=110, d=111

然后数一下就行了。

1.6 D 这道题我是穷举的 orz.....一共这么几种情况：

118,127,136,145;

226,235,244;

334; 然后有数字重复的算 3 种排列，不重复的算 6 种排列，共计 $4 \times 3 + 4 \times 6 = 36$ 种。

1.7 D 这题很基本了。

1.8 D 一般学过操作系统这门课的都会吧，而且个人觉得 D 这个选项的出现不符合 Google 风格。

1.9 D 这题其实很好做，因为 D 肯定是对的，而且 ABC 的言论太绝对。但如果一定要给出解释的话.....

A 选项的优化只能针对代码本身，纯系统调用什么的是不会性能提升的（当然也不会下降），

B 选项我觉得是在并行优化方面，好的编译器可以从循环中发掘并行性，展开之后就不行了，

C 选项有点说不清。消除数据依赖主要有两个方法，一种是 SSA，即静态单赋值[3]，这是通过对变量进行重命名实现的，严格的说应该叫“寄存器重命名”[4]而不是“寄存器分配”；另外一种调换指令顺序，这种只要不是真相关（写后读，RAW）的话都可以消除掉，也不属于寄存器分配。所以感觉不应该选这个。

1.10 B 求最大公约数用的是辗转相除法（欧几里得算法），所以是 $O(\log n)$ [5]。

2.1 这题比较基本，而且很多企业的笔试都爱考类似的。主要就是对尝试对数 a 进行质因数分解，最容易写的就是从 2 开始一直除到 \sqrt{a} ，性能提升一点就从 2,3 然后除奇数一直到 \sqrt{a} 。当然还可以优化一下，建立一个动态质数链表，将之前取到的所有质数加入表进行加速。

2.2 这题我觉得除了重载一下 swap 函数然后用传统排序法之外也想不出什么高效的办法了。而且要代码实现，时间紧迫也不由得你多想。







2.3 这题个人觉得是这场笔试唯一拉区分度的题了（所以非科班出身的本人妥妥的死在这道题上），基于动态规划算法。事实上就是写出 LD 算法的伪代码，[6]中有详细的描述。

考场里完全对这东东没概念，就随便写了点啥交掉了。好吧，目送各位进面试的大牛顺利（你们的考验才刚刚开始什么的我会随便乱说吗）

2.3 GOOGLE 中山大学笔试归来

本文原发于应届生 BBS，发布时间：2012 年 9 月 24 日

地址：<http://bbs.yingjiesheng.com/thread-1544768-1-1.html>

选择题亚历山大  最讨厌操作系统了  编程题感觉还不错  和华理几个同学讨论下说我第二题算法很不错   最后一道大题算法题木有时间做  会悲剧么 坐等大神

2.4 2010.09.27-谷歌笔试题详尽版

本文原发于应届生 BBS，发布时间：2010 年 9 月 28 日

地址：<http://bbs.yingjiesheng.com/forum.php?mod=viewthread&tid=698129>

前面的童鞋发的不够详尽，这个是我回忆+记录的，尤其是选择题，包括选项都写了，大家一起讨论下吧，顺便攒RP~



[20100927 GOOGLE 笔试.doc](#) (37.5 KB, 下载次数: 241)

1.

哪个表达式不能用这个匹配: $a(bc)^*d$?

A.ad

B. abcd

C.abc

D.abccd

2.

INTEL X86 CPU 中，哪种运算最慢：

A. 加

B. 减

C.乘

D.除

3.

下面程序的输出：

```
Fun(){
bool first =true;
int sum = 0;
int cur;
for(unsignedshort i=65535; i>=0; --i){

if(first){

cur=65536;

sum+=cur%3;

first=false;
}else{

sum+=--cur%3;

if(cur<=0)

printf(“%d,%d”, sum, i)break;

}

}

}
```

- A.
65535, 0
B.65536,1
C.65536,65535
D.65536,0
4.
有 19 本书，分别编号为 1-19，从中选出 5 本，要求任意两本不相邻，一共有多少种选法？
A. 2002
B. 3003
C.11628
D.360360
5.
一套房子 200 万，每年价格上涨 10% ，一个工程师每年固定收入 40 万，假定他不贷款，不涨工资，问几年能买的起房子
A. 5
B.7
C.8
D.永远也买不起
6.
有 N 个叶节点的满二叉树节点，其共有多少个节点？
A. $2N-1$
B. $2N$
C. $N-1$
D. N
7.
以下哪个排序算法的最坏时间复杂度是 $O(n\log n)$?
A.
归并排序
B. 快速排序
C. 冒泡排序
D.插入排序
8.
两个排好序的数组大小为 N,M，合并成一个有序数组，则最小比较次数：
A. $\min(N,M)$
B. $M+N-1$
C. $N+M$
D. $\max(N,M)$
9.
关于 TLB 和 Cache，下面哪个说法是错的
A.
TLB 和 cache 中存的数据不同
B.
TLB miss 后，可能在 Cache 中直接找到页表内容
C.

TLB miss 会造成程序执行出错，但是 cachemiss 不会

D.

这两者的命中率都与访存模式有关

10.

对于数据库，以下哪种说法是错的

A.

每个表都必须有主键

B.

跨表查询很慢

C.

数据库不支持多个客户端同时对一个表进行写操作

D.

多维索引可以用 KD 树

编程题（前两个写程序，最后一个写思路或者伪代码）

1.

用一个数组 $A[N+1]$ 存储一个多项式： $a_0+a_1x+a_2x^2+\dots+anx^n$ ，用一个程序计算这个多项式的值。

函数原型：`double eval(double x, double *A)`

2.

有 n 个队伍， $n=2^k$ 。有一个二维数组，`winner[i][j]` 代表第 i 队和第 j 队的比赛结果中胜出队伍的编号，`winner[i][j]` 和 `winner[j][i]` 相同。另有一个代表单淘汰赛签位的一维数组 `order[0]...[n-1]`，`order[i]` 代表 i 签位上的队伍编号。现在要求输出一个最终队伍排名，如果在同一轮中淘汰的认为排名相同，并且时间和空间复杂度尽可能低

如 $n=4$ 时有一个例子（例子不记得了）

函数原型：`void fun(int **winner, int *order, int *result)`

$0 < n \leq 1000$

3.

KOF 里的连招。连招表达式 $S \rightarrow T$ ，比如 $ABC \rightarrow C, ABD \rightarrow E, BDE \rightarrow F, DEF \rightarrow G$ ，那么连招输出就可以是 $ABD \rightarrow E \rightarrow F \rightarrow G$ 。现在要求一个程序，能够输出最大连招的长度

判卷准则：

1.

前 10 个小题答对了至少 6 个才会判后面的大题

2.

大题最低分数为 20（每题 10 分），需满足其最低分数线。

2.5 Google 笔试归来

昨天刚参加 Google 宣讲和笔试，考得很基础，共享一下笔试题目，奇文共赏之。顺便攒 RP，哈哈 10 道选择题，3 道大题。10 道选择对六道才看你答题，否则直接被 BS。

1. 考的是正则表达式，什么字符串匹配，没看过，:87)~~。

2. 在 Intel 8086 中，加减乘除那个整数运算最耗时。很基础哇~~

3. 看程序，写算法，考察的是 `unsigned short` 类型的范围。程序有点长，变量名还相似，想不起来了，

4.19 本书，编号从 1-19。从中抽五本，任意相邻两本不是相邻编号的情况有多少种。这个题谁会啊，大家发帖探讨一下。

5.N 为满二叉树的叶子节点数，求总结点数。确实很基础~~

6.排序算法：在最坏情况下时间复杂度为 $O(n \log n)$ 的是归并，快速，冒泡，插入中的哪个。

7.房价 200 万，每年以 10% 的速度递增，工程师为 40 万年薪，问什么时候买得起房。

8.有两个有序数组长度为 M 和 N，将两个数组合并，最好情况下比较几次。M 次，N 次， $\min(M,N)$, $\max(M,N)$

9.TLB 和 Cache 的区别，这个题不会，没听说过 TLB。上网查了查，TLB: Translation lookaside buffer, 即旁路转换缓冲，或称为页表缓冲；里面存放的是一些页表文件（虚拟地址到物理地址的转换表）。大家还是自己上网了解吧。

10.数据库的试题，偶记不清了，不过不难。

一 写函数 `double value(double x, double A[], double N)` `double A[N]` 存储多项式 $f(x) = a_0 + a_1x + a_2x^2 + \dots$ 的系数。N 为已知。

二 有 2^K 队伍比赛，按照 order 给出一个比赛顺序的排列，order 表示编号为 i 的队的位置，呀呀，记不太清了，有 `winner[j]` 表示 i, j 两队比赛结果，只有胜负没有平局，`winner[j] = winner[j]` 求 `result[]` 里面存放各队比赛排名。貌似用递归，只是小弟拙见，哈哈，

三 KOF 里连招，简化为 ABCD.....Z 求最长连招..... 记不清了，见谅，XDJM 补充。有答案飘过来啊

2.6 google 北大笔试题

1. 1 关于 IP 协议那个正确

A IP 是 TCP 上层协议 B IP 协议是应用层协议 C 由于两个属于同一层协议，他们之间可以直接通信 D IP 协议不提供可靠的通信

1.2 关于内存正确的是

A 内存的存取速度不能低于 cpu 速度，否则会造成数据丢失

B 程序只有在数据和代码等被调入内存后才能运行

C 采用虚拟内存技术后程序可以在硬盘上直接运行

D 某计算机的内存容量为 16MB，那么他的地址总线为 24 位

1.3 单链表中结点的结构为 (data, link)，若想删除结点 p（不是头节点或者尾结点）的直接后继，则应执行下列哪个操作

A `p->link ; p->link=p->link->link` B `p->link->link=p->link`; C `p=p->link->link` D `p->link=p->link->link`

1.4 已知 $x \geq y$ and $y \geq z$ 为真，那么 $x \geq z$ or $y = z$ 值为

A 真 B 假 C 无法确定 D x y z 同为正数时为真

1.5 某请求被随即分配到四台机器进行处理，分配到每台机器的概率 A 15% B 20% C 30% D 35%，处理请求的失败概率分别为 5% ， 4%， 3% 2%，现在请求失败，问由 C 造成的概率最接近 A 26% B 28% C 30% D 32%

1.6 假设我们用 $d = (a_1, a_2, \dots, a_5)$ 表示无向无环图 G 的 5 个顶点的度数，下面给出的哪组值是可能的

A {3,4,4,3,1} B {4,2,2,1,1} C {3,3,3,2,2} D {3,4,3,2,1}

1.7 设栈 S 和队列 Q 的初始状态为空，元素 $e_1, e_2, e_3, e_4, e_5, e_6$ 一次压入栈 S，一个元素出栈后即进入队列 Q，若出队列的顺序为 $e_2, e_4, e_3, e_6, e_5, e_1$ 则栈 S 的容量要求最小值为

A 2 B 3 C 4 D 5

1.8 在堆排序算法中我们用一个数组 A 来模拟二叉树 T，如果该 A[0]存放的是 T 的根节点，那么 A[K](K>0)的父亲节点是

A $(K-1)/2$ B $K/2$ C $(K+1)/2$ D 都不对

1.9 现有如下任务需要安排在若干机器上并行完成，每个任务都有开始时间和结束时间（开始和结束时间都包括在任务执行时间内）的要求

任务名称	开始时间	结束时间
a	1	7
b	8	9
c	2	5
d	7	11
e	3	6
f	7	9
g	10	13

则最少需要使用的机器数目为

A1B2C3D4

1.10 在设计一个操作系统时，哪项不是必须考虑的

A 设备管理模块 B 文件系统模块 C 用户管理模块 D 进程管理模块

2.1 正整数序列 Q 中的每个元素都至少能被正整数 a 和 b 中的一个整除，现给定 a 和 b，需要计算出 Q 中的前几项，例如，当 a=3，b=5，N=6 时，序列为 3，5，6，9，10，12

(1) 设计一个函数 void generate (int a,int b,int N,int * Q) 计算 Q 的前几项

(2) 设计测试数据来验证函数程序在各种输入下的正确性

2.2 有一个由大小写组成的字符串，现在需要对他进行修改，将其中的所有小写字母排在答谢字母的前面（大写或小写字母之间不要求保持原来次序），如有可能尽量选择时间和空间效率高的算法 c 语言函数原型 void proc (char *str) 也可以采用你自己熟悉的语言

2.3 已知一颗无向无环连通图 T 的所有顶点和边的信息，现需要将其转换为一棵树，要求树的深度最小，请设计一个算法找到所有满足要求的树的根结点，并分析时空复杂度（描述算法即可，无需代码）

2.7 google 浙大笔试题

一、单选

1、80x86 中，十进制数-3 用 16 位二进制数表示为？

2、假定符号-、*、\$分别代表减法、乘法和指数运算，且

1)三个运算符优先级顺序是：-最高，*其次，\$最低；

2)运算符运算时为左结合。请计算 $3-2*4\$1*2\3 的值：

(A)4096, (B)-61, (C)64, (D)-80, (E)512

3、下列伪代码中，参数是引用传递，结果是？

```
calc(double p, double q, double r)
{q=q-1.0;r=r+p}
main(){
double a = 2.5, b = 9.0;
calc(b-a, a, a);
```

```
print(a);  
}
```

(A)1.5 (B)2.5 (C)10.5 (D)8 (E)6.5

4、求输出结果:

```
int foo(int x, int y){  
if(x <=0 || y <= 0) return 1;  
return 3 * foo(x - 1, y / 2);  
}
```

```
printf("%d\n", foo(3, 5));
```

(A)81 (B)27 (C)9 (D)3 (E)1

5、下列哪个数据结构在优先队列中被最广泛使用?

(A)堆 (B)数组 (C)双向链表 (D)图 (E)向量

6、以下算法描述了一个在 n 个元素的双向链表中找到第 k 个元素的方法 ($k \geq 1$ 且 $k \leq n$):

如果 $k \leq n - k$, 从链表开始往前进 $k-1$ 个元素。

否则, 从终点出发, 往回走 $n - k$ 个元素。

这个算法的时间代价是?

(A) $\theta(n \log n)$ (B) $\theta(\max\{k, n - k\})$ (C) $\theta(k + (n - k))$

(D) $\theta(\max\{k, k - n\})$ (E) $\theta(\min\{k, n - k\})$

7、有一个由 10 个顶点组成的图, 每个顶点有 6 个度, 那么这个图有几条边?

(A)60 (B)30 (C)20 (D)80 (E)90

8、正则表达式 $L = x^*(x|yx)^+$ 。下列哪个字符串不符号 L

(A)x (B)xyxyx (C)xyx (D)yxx (E)yx

9、为读取一块数据而准备磁盘驱动器的总时间包括

(A)等待时间 (B)寻道时间 (C)传输时间 (D)等待时间加寻道时间

(E)等待时间加寻道时间加传输时间

二、算法

1、打印出一个二叉树的内容。

2、在一个字符串中找到第一个只出现一次的字符。如 abaccdeff, 输出 b。

3、给定一个长度为 N 的整数数组 (元素有正有负), 求所有元素之和最大的一个子数组。分析算法时空复杂度。不必写代码。

2.8 google 交大笔试题

在宣讲会的时候, 听旁边的师姐说上海只招两个职位每个职位只招一个人。现在后悔只选了北京和上海的 SWE 了。

不过反正……也不指望了。。

笔试题目：9 道单选+3 道问答

时间：100 分钟

我做的是 B 卷。

单选题：

1，求两个二进制数的异或值，基本上学过一点计算机的东西的人都能对的题目。。

2，不记得了。。也是不需要思考的题目。。

3，大概是如下的函数：

```
int someFunc(int x){
if (x == 0)
return 0;
else
return x + someFunc(x - 1);
}
```

问这个计算的是什么。。

4，不记得了。。不需要思考吧。。

5，不记得了。。不需要思考吧。。

6，参见 2，4，5。。

7，似乎需要思考一下。。

8，问链表结构和数组相比的优势不包括哪项，

包括：

插入的时间

删除的时间

存储空间

剩下两个不记得了。。

9，如下函数：

$$T(x) = 1 \quad (x \leq 1)$$
$$T(n) = 25 T(n/5) + n^2$$

问 $T(n)$ 随 n 的增长。

选项大概是这样的：

$O(n^2)$ ， $O(n^2 \log n)$ 等等的。。

问答：

1，写两个 $N * N$ 的矩阵的乘法，给出了 C 的格式，你可以选择你喜欢的语言去写。。

```
int* multi(int* a1, int* a2, int N){
```

```
}
```

2，寻找一个单向链表的中项，如果存在两个则返回前一个。给出了 C 的格式，同样你可以选择。。。。

```
struct {
```

```
Node* next;
```

```
int value;
```

```
} Node;
```

```
Node* someFunc(Node* head){
```

```
}
3, 给一个长度为 n 的整数数组, 只允许用乘法不允许用除法, 计算任意(n-1)个数的组合乘积中最大的一组。。。写出算法的时空复杂度。
ps: 怀疑这道题目出错啦。。虽然我也做错了。。。。。。
```

一些补充:

- 1, 问答的第一题是 google 上学期 intern 的大题原题;
- 2, google 很喜欢考链表, 无论 intern 的面试以及两次的笔试都有这样的题目;
- 3, google 一般大题第三道都是写算法的时空复杂度;
- 4, 选择题基本上偏简单, 但是要做得准确率高似乎并不那么容易;
- 5, 根据传言, 小道消息, 人云亦云以及以讹传讹, google 的高速审卷政策来源于审卷时以选择题为主, 如果你全对啦, 那么恭喜你 pass 啦; 如果你错了好几道, 那么下次努力吧, 如果还有下次。。。大题基本是做参考的。。。
- 6, 选择题很多记不清了, 因为一遍做下来的, 回去随便扫了两眼。。。加上过了这几个小时, 记不得了。希望大家补充修正以及修改。。。

2.9 google 笔试集锦

选择题+三道算法题

选择题没什么难的 最后一道考的数据库使用什么存储结构不会做。。

算法题

第一题没什么好说

第二题可破坏一个数组 $A[0..N-1]$ 的条件下使用最少的内存判断是否存在相同的元素

我的做法是堆排序 时间 $O(N\log N)$ 空间 $O(1)$ 复杂度上来看应该最优了

第三题已知每个点的父节点, 求这棵树的最大独立集

用递归求解 类似动态规划 但是不存在重叠子状态 经典算法问题了

预处理每个节点的子节点存在一张表里

时间 $O(N)$ 空间 $O(N)$

大家做的结果是这样吗?

在宣讲会的时候, 听旁边的师姐说上海只招两个职位每个职位只招一个人。

现在后悔只选了北京和上海的 SWE 了。

不过反正……也不指望了。。。

笔试题目: 9 道单选+3 道问答

时间: 100 分钟

我做的是 B 卷。

单选题:

- 1, 求两个二进制数的异或值, 基本上学过一点计算机的东西的人都能对的题目。。
- 2, 不记得了。。也是不需要思考的题目。。
- 3, 大概是如下的函数:

```
int someFunc(int x){
    if (x == 0)
```

```
    return 0;
else
    return x + someFunc(x - 1);
}
```

问这个计算的是什么。。。

- 4, 不记得了。。不需要思考吧。。
- 5, 不记得了。。不需要思考吧。。
- 6, 参见 2, 4, 5。。
- 7, 似乎需要思考一下。。
- 8, 问链表结构和数组相比的优势不包括哪项,

包括:

插入的时间

删除的时间

存储空间

剩下两个不记得了。。

9, 如下函数:

$T(x) = 1 \ (x \leq 1)$

$T(n) = 25 T(n/5) + n^2$

问 $T(n)$ 随 n 的增长。

选项大概是这样的:

$O(n^2)$, $O(n^2 \log n)$ 等等的。。

问答:

1, 写两个 $N \times N$ 的矩阵的乘法, 给出了 C 的格式, 你可以选择你喜欢的语言去写。。

```
int* multi(int* a1, int* a2, int N){
}
```

2, 寻找一个单向链表的中项, 如果存在两个则返回前一个。给出了 C 的格式, 同样你可以选择。。。

```
struct {
    Node* next;
    int value;
} Node;
Node* someFunc(Node* head){
}
```

3, 给一个长度为 n 的整数数组, 只允许用乘法不允许用除法, 计算任意 $(n-1)$ 个数的组合乘积中最大的一组。。写出算法的时空复杂度。

ps: 怀疑这道题目出错啦。。虽然我也做错了。。。。。。

一些补充:

- 1, 问答的第一题是 google 上学期 intern 的大题原题;
- 2, google 很喜欢考链表, 无论 intern 的面试以及两次的笔试都有这样的题目;
- 3, google 一般大题第三道都是写算法的时空复杂度;
- 4, 选择题基本上偏简单, 但是要做得准确率高似乎并不那么容易;

5, 根据传言, 小道消息, 人云亦云以及以讹传讹, google 的高速审卷政策来源于审卷时以选择题为主, 如果你全对啦, 那么恭喜你 pass 啦; 如果你错了好几道, 那么下次努力吧, 如果还有下次。。。大题基本是做参考的。。。

6, 选择题很多记不清了, 因为一遍做下来的, 回去随便扫了两眼。。。加上过了这几个小时, 记不得了。希望大家补充修正以及修改。。。

7, google 会在 11 号开始 3 天内发面试通知, 据小道消息等等, 有四轮面试。bless 大家~~

输入 $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$, 如何在 $O(n)$ 的时间, 用 $O(1)$ 的空间, 将这个序列顺序改为 $a_1, b_1, \dots, a_n, b_n$ 。

这个问题在瀚海星云上跟出了好几十的跟帖, 看看大家有没有什么好的解法!!!

晚上是 google 的校园宣讲会

先前并没有投简历, 但是还是奔了去

google 的 hrjj 看起来仿佛没有 ibm 的那么动人, 但却很亲切

五点半左右开始, 开始前那个 hrjj 放了些 google 员工自己拍的 mv,

比较有意思

之后是一个做技术的男生如数家珍的介绍 google 做的东西

之所以称为“男生”主要感觉他很学生气。。。讲话并没有那些很商业的调度气氛的东西

很理想主义, 呵呵, 虽然听的很多人打瞌睡

他总是说 google 是个很理想主义的公司, 虽然为了生存不得不做一些商业化的事情, 但是, google 做很多事情都是因为认为觉得有做的价值并且要做好, 所以有了 googleprint, googleearth, 等等

不过, google 仿佛是一个需要聪明人的公司, 接下来的笔试应证了这一点

内容很少, 甚至简单, 选择题可能是送分的, 大概 10 来道, 并不难, 算些东西, 还有一点点程序方面的基础概念, 后面是三个算法设计题。

第一个, 深拷贝一个二叉树。我不明白这道题的动机是什么, 我最后很搓得用了递归, 虽然明知道这样很耗, 可是实在想不起来非递归算法怎么个写法了。

第二个, 把输入数组随机分配到一个新的数组上, 每个数都完全对应一个随机的位置, 当然, 随即产生函数是提供了的。这个题我想了很久, 冲突的时候怎么办? 散列? 但是那样还是随机的吗?

第三个, 很 ft。居然是 C 语言的一个作业题。N 个人排成圈, 从第一个人开始, 去掉, 隔一个人, 去掉下一个, 以此类推, 要求出最后出局的那个人的位置。并且, 要求分析算法的时间、空间复杂度。我感觉这个题表面简单, 实际对算法的优化要求很高。一个是空间复杂度, 一个是时间复杂度, 只是不知道做到什么程度算是极好。

仿佛这种类型的考试, 大家差不多都能答完, 只是如何才能与众不同却让人伤脑筋

回来后对 google 的印象变得很好, 至少感觉他们真的是需要人, 而不是一个螺丝钉。。。

昨天, 参加了 Google 春季实习工程师的招聘笔试。这是进入大学以来参加的第一次笔试。

可以说, 这次笔试也让我更明确了我的学习方向。学习什么技能不重要, 重要的是要有扎实基础, 数据结构, c++ 语言等, 都要扎实的掌握, 不可有丝毫的马虎。于是, 渐渐开始后悔了, 后悔自己对基础知识没有学好, 却又一天到晚想着要学什么什么东西, 这真是丢了西瓜捡了芝麻, 一点便宜都没有占到啊。

还记得昨天的笔试, 第一题是有关位操作的, 很简单。接下来两题是讲 const 指针的, 这便是我的一个薄弱点, 以前一直都没有学好。首先, `const char *p="abcde"`, 那么, p 指向的内容是不可变得, 即不能改变 *p 的值, 但是, p 却是可变的, 即可以这么操作: `p="12345"`, 或者 `p=new char[10]`, 等等; 另外, `char *const p="abcde"`, 则表示这是一个指针常量, 即 p 不能指向另外的地址, 但是, p 指向的内容是可以改变的; 还有一个就是常量指针常量,

`constchar*constp`，它是指向常量的常量指针，若初始化的时候，`p` 指向的是一个变量，那么，不能用指针操作来改变变量的值，例如：`inta=0;constint*constp=&a;*p=1;`这句或是错误的，但是，可以直接给变量赋值，即 `a=1;` 是可以的。

接下来几题涉及了算法分析的一些东西，然后是涉及语法分析的正则表达式什么的，这个因为没学过编译，也就不太懂了。接下来是考了抖动。什么是抖动？所谓抖动，主要是由于页面交换过于频繁，而导致时间浪费太多的现象。特别的，在虚拟内存技术下，若工作集大小选的不合适的话，就会有频繁的换入换出，而我们知道，硬盘的读写是需要花费大量时间的，这样，就导致了某进程等待的时间比实际运行的时间多的多。

接下来的大题是编程题和算法题，前两题都挺简单的，第一题要求完成双向链表的添加结点，第二题要求比较两字符串的字符出现频率；第三题是个算法题，基本意思很简单，但是，要找到一个好的算法却需要一定的思考能力，具体题目已经记不太清了。

总之，这次笔试，虽然说从一开始所抱的心态就是见识见识，没有奢求什么，只是想走出这个温暖的象牙塔，走到外面这个花花世界来看看情形。一切都需放开。走了一遭才发现，不要总往高处看，基础扎实了，才能造起巴比伦塔。

1.写程序判断是否字符串 A 里每个字符在 A 中出现的次数都大于在字符串 B 中出现的次数。

注：此题我是对每个字符出现的次数分别统计，然后比较。重复的字符重复统计比较，所以效率很低。不知有什么好的改进方法？

2.对一个数组 S，求其中满足要求的最大元素 C。要求 C 满足等式 $C=A*B$ ，其中 A、B 也是数组 S 中的元素。请用能想到的最优算法，分析时间和空间复杂度。（用语言描述算法，不用写程序）

注：这个题我当时做的方法在时间上要用 $O(n^3)$ ，事后想出了个 $O(n^2 \lg n)$ 的方法。不知有没有更好的方法。

传说中有人说：google 笔试很变态，题目稀奇古怪；google 题目比较基础简单。

终于得于一见。

晚上干完活才去，到的时候已经 7 点左右。在二教已经没有空余座位，无赖多等了 20 多分钟才安排了一个新的教室。

发题，答卷。

整体感觉就是比较基础，没有什么很难的题目，除最后一题目以外。

选择题目 9 个，比较基础。

第一题，相对比较简单，递归

第二题，基本是看是否对函数效率是否有概念。似乎也跟动态规划的记录法有点相似之处。

最后一题目回想起来实在惭愧，其实还是题目没有怎么读懂。嗨！

第一和第二应该是不错的选择！

特别是第二种方法！

也晚了，干脆再折腾一会儿。

题目： n 个点组成的无向图，对于任意给出的两点，判断其是否有长度为 K 的通路。

分析：无向图是允许有环存在的，也允许两点间有多条边，长度为 K 是指从一个点移动到相邻点这样过程的次数。比如 AB 两点相邻，那么 A-B-A-B 就是长度为 3 的通路。

解法一：直接利用图论中的一个定理，邻接矩阵 U 的 K 次方后所得矩阵的 (i,j) 元素即为原无向图中 i,j 两点之间长度为 K 的通路数量。因此复杂度只需要对 U^K 分析。

由于 U^2 的时间复杂度为 $O(n^3)$ ，（这里使用了传统矩阵乘法，不过即使用 strassen

方法也依然是 $O(n^{2+e})$ ），其中 0 小技巧，不需要去依次乘，否则的话需要 $K-1$ 次矩阵乘法，可以从 U^2 ，算出 U^4 ，

依此类推，因此总的时间复杂度为 $O(n^3 * \lg K)$ 。空间复杂度不细述 $O(n^2)$ 。

评论一：细心的人会注意到这样的解法其实比题目的要求要强，因为题目中只是要求出任意给定的两点间 K 通路数目，而上述做法，则是把所有的两点组合都求出来了。因此还应该存在着改进的算法在 $O(n^2)$ 这个时间复杂度级别。

解法二：这是看水木上一个网友的解法，我重写成下面这样：设所给的两点为 A 、 B 。那么首先把 A 的所有相邻点全部放入 U 集合中，接着对之前放入 U 中的点 i 分别求出其相邻点，并将它们放入 U 中（而原先的点 i 本身则从集合中事先去除）。这样做过 K 次后，看队列中是否有 B 点。若有则表明存在着长度为 K 的通路。由于每次寻找邻接点的次数不超过 n^2 ，而要进行 K 次，因此时间复杂度为 $O(n^2 * K)$ ，空间复杂度为 $O(n)$ 。

评论二：和解法一相比，从 $n^3 * \lg K$ 变为了 $n^2 * K$ ，也就是 $n * \lg K$ 与 K 之间的差别，若 $n=k$ 的话，也就说明后者比前者快了 $\lg K$ 倍。

解法三：从 A 到 B 利用深度优先遍历，找出每一条 AB 间的路径，其中没有点重复经过。

对每一条这样的路径进行分析：

若长度 $> K$ ，该路径不可能满足；

若长度 $= K$ ，则已满足；

若长度一次加一)。若无环，则看 K -当前路径长度，若该值为偶数，则可以在该路径上的某两个邻接点上来回的重复走从而满足要求（每来回走一次加二）。其他情况，无法满足。

由上得到算法的时间和空间复杂度均为 $O(n^2)$ ，

评论三：时间复杂度很可能不是 $O(n^2)$ ，因为从 A 到 B 上找出每一条 AB 间的简单路径，由于存在着重复边和环的情况，多半就不是简单的深度遍历的 $O(n^2)$ 了，具体是多少我也说不清。笔试里我用的这个方法，现在感觉挺玄。。。

2.10 google 的 21 道 GLAT 考试

10 月底，Google 在美国《麻省技术评论》、《LinuxJournal》、《Mensa》、《今日物理》等几本专业杂志上，刊登了一份"Google 实验室能力倾向测试"。

试卷开头，蛊惑地写着"试试看！把答案寄回 Google，你有希望去 Google 总部参观，并成为我们其中一员"。

1. Solve this cryptic equation, realizing of course that values for M and E could be interchanged. No leading zeros are allowed.

WWWDOT - GOOGLE = DOTCOM

2. Write a haiku describing possible methods for predicting search traffic seasonality.

3.

1

1 1

2 1

1 2 1 1

1 1 1 2 2 1

What is the next line?

4. You are in a maze of twisty little passages, all alike. There is a dusty laptop here with a weak wireless connection. There are dull, lifeless gnomes strolling about. What dost thou do?

- A) Wander aimlessly, bumping into obstacles until you are eaten by a grue.
- B) Use the laptop as a digging device to tunnel to the next level.
- C) Play MPoRPG until the battery dies along with your hopes.
- D) Use the computer to map the nodes of the maze and discover an exit path.
- E) Email your resume to Google, tell the lead gnome you quit and find yourself in whole different world.

5. What's broken with Unix?
How would you fix it?

6. On your first day at Google, you discover that your cubicle mate wrote the textbook you used as a primary resource in your first year of graduate school. Do you:

- A) Fawn obsequiously and ask if you can have an autograph.
- B) Sit perfectly still and use only soft keystrokes to avoid disturbing her concentration.
- C) Leave her daily offerings of granola and English toffee from the food bins.
- D) Quote your favorite formula from the textbook and explain how it's now your mantra.
- E) Show her how example 17b could have been solved with 34 fewer lines of code.

7. Which of the following expresses Google's over-arching philosophy?

- A) "I'm feeling lucky"
- B) "Don't be evil"
- C) "Oh, I already fixed that"
- D) "You should never be more than 50 feet from food"
- E) All of the above

8. How many different ways can you color an icosahedron with one of three colors on each face?

What colors would you choose?

9. This space left intentionally blank. Please fill it with something that improves upon emptiness.

10. On an infinite, two-dimensional, rectangular lattice of 1-ohm resistors, what is the resistance between two nodes that are a knight's move away?

11. It's 2 PM on a sunny Sunday afternoon in the Bay Area. You're minutes from the Pacific Ocean, redwood forest hiking trails and world class cultural attractions. What do you do?

12. In your opinion, what is the most beautiful math equation ever derived?

13. Which of the following is NOT an actual interest group formed by Google employees?

- A. Women's basketball
- B. Buffy fans
- C. Cricketeers
- D. Nobel winners
- E. Wine club

14. What will be the next great improvement in search technology?

15. What is the optimal size of a project team, above which additional members do not contribute productivity equivalent to the

2.11 Google 惯用“整蛊题”

Google 上一轮招聘，今年夏天刚结束。

用的也是一道“科学麻瓜”看不懂的“整蛊题”，而且，堂而皇之挂在硅谷各大地铁站上。9 月底，3 块 15 米长的米色广告牌上，简简单单刷着“(在‘e’的数列中所能找到的第一个十位数质数).com”，没有公司名也没有任何广告词。

花了几秒钟，路人才明白，这是一道数学题。自然常数 e (2.718281828……) 的第一个十位数质数，是目标网站的名字。

好奇分子忍不住用 Google 搜索起答案来，压根儿不晓得这就是 Google 出的“硬骨头”考题。

不少人后来在规定时间内，登录上了 www.7427466391.com。然而，那不是梦寐以求的终点站，Google 恶作剧似的，为“高手”们在半山腰设了个休息的小凉亭。

www.7427466391.com 里，贴出一条更令人头疼的数学问题，答出这个问题，能得到进入下一个网页的密码。

跑完数学“马拉松”，7500 个“幸存者”走入 Google 实验室网页，成功投出简历。最后，Google 只要了 50 个人。“光以广告而论，Google 也算得上高段！”

波士顿一家广告公司的高级副总裁弗里茨·库恩分析，“目标人群看到广告后会想，‘这是我的语言，那是冲着我来’；对其他人而言，广告也使 Google 的形象大大提升。他们可能会想，‘我是得不到这份工作的了。不过，在那儿工作的人真聪明’。”

Google 测试考的就是脑筋

• 试着证明 $WWW.DOT.GOOGLE=DOT.COM$ • 用俳句（一种日本短诗，每句有一个与季节有关的词）来描述各种模型，借此预测网络搜索流量的季节性变化。

• 你落入一个迷宫，回旋不断的走廊。手里有一台堆满灰尘的手提电脑，可以无线上网。周围，许多无生命的侏儒徘徊走动。这种情况下，你会如何做？

A) 无目的地徘徊，不停走入死胡同，然后被迷宫里面的妖怪吃掉。

B) 用手提电脑当铲子，打穿地板直接进入游戏下一关。

C) 玩网络游戏《魔法奇兵》，直到电池耗尽。

D) 利用计算机，找到迷宫的节点，发现准确出路。

E) 把你的简历寄给 Google，告诉迷宫里领头的妖怪，你要退出游戏。然后，发现你回到了现实世界。

• Unix 有什么问题？你会如何补救它？

• 你在 Google 工作的第一天，发现你同寝室的室友，曾写过一本书。你研究生一年级时，这本书是你最重要的参考资料。你会：A)求他帮你签个名。B)不改坐姿，却放轻打字声音，尽量避免影响他。

C)把你每天吃的麦片和咖啡，留给他吃。

D)引用他那本书中间，你最喜欢的程式，告诉他这则方程给了你多少启发。

E)让他看看，你可以用不到 34 句语句，完成一个高难度程序。

• 以下哪个最好地表达了 Google 的企业文化？

A) “我感觉挺幸运”

B) “别干坏事”

C) “哦，我已经完成了任务”

D) “你身边 10 米以内，必定能找到食物”

E)以上皆是 • 用 1 欧姆的电阻，组成无限大的放行点阵，问“象棋跳马步”（“日”字对角点）两点之间的电阻是多少？

• 下午 2 点，旧金山著名的湾区。你可以选择去阳光海岸、国家公园的红杉林里徒步旅行，或者参观城市里的文化景观。你会怎么做？

• 搜索技术的下一个革命性突破是什么？

• 一个技术研究小组的最优化人员组合是几个人？一旦超过这个数字，每增加一个研究员，平均生产力就会相应下降：A)1B)3C)5D)11E)24 • 三角形 ABC，用圆规和尺，找出点 P，保证三角形 ABP、ACP 和 BCP 周长相等。

• 你写过最酷的程序是什么？

• 找出此数列的下一个:10,9,60,90,70,66?A)96B)10 的 100 次方 C)A 或者 BD)以上皆否 • 用少于 29 个词，描述你能带给 Google 实验室的贡献。

“微软是个公平的公司，这里几乎没有特权。盖茨只是这两年才有了自己的一个停车位。以前他来晚了没地儿，就得自己到处去找停车位。”

“微软非常强调员工的动手能力。在做新产品发布时，盖茨都能自己动手做演示。他总是在和工程师作搭档，对自己的产品很熟悉，这样，任何人都糊弄不了他。”

1.单项选择题

1. 下面一段代码的输出是[]

```
void fn( int* b){
    (*b)++;
}
int main(){
    int a=7;
```

```

    fn(&a);
cout<<a;
return 0;
}

```

A.0 B.7 C.8 D.undefined

2. 定义 int i,j,*p=&i; 那么下面哪条语句可以完成 i=j 的赋值[]

A.i=*p; B.*p=*&j; C.i=&j; D.I=**p;

3. 用二叉搜索树和哈希表存储相同的数据集,对于以下何种操作,二叉搜索树比哈希表速度更快?[]

A.检索 B.插入 C.删除 D.更新 E.排序

4. 包含 N 个几点和 M 条边的有向带权图 G, 边的权为正, 以下操作中不可以在 O(N+M) 的时间复杂度内完成的操作是:[]

- A. 求结点 s 到结点 t 之间的最短距离
- B. 求距离结点 s 最近的结点
- C. 已知起始结点, 对图 G 中的结点进行拓扑排序
- D. 求图 G 的最大强连通子图

5. 有如下递归函数 f(n),其时间复杂度为[]

```

int f(int n){
    if(n==0)
        return 0;
    if(n==1)
        return 1;
    return ( 5*f(n-1) - 6*f(n-2));
}

```

A.O(n) B. O(n^2) C. O(n^3) D. O(2^n)

6. 下面所述步骤中,哪一个不是创建经常所必需有的[]

- A.由调度程序为进程分配 CPU
- B.建立一个进程控制块
- C.为进程分配内存
- D.将进程控制块链入就绪队列

7. 在多进程的系统中,为了保证公区变量的完整性,各进程应互斥进入临界区。所谓临界区是[]

- A.一个缓冲区
- B.一个数据区
- C.一个同步机构
- D.一段程序

8. 能产生满足如下条件语言的正则表达式是: 1.每一个 a 后至少紧跟两个 c; 2.每一个 b 后至少紧跟一个 c []

A.(acc|bc|c)* B.(acc|bc)* C.(ac|bc)* D.不是正则语言

9. 以下哪项不是 RPC (远程过程调用) 的特点[]

- A.速度快
- B.降低系统耦合度
- C.可以实现异构系统间的协作

10. 有三个桶, 容量分别是 3 升, 5 升, 7 升, 你只能进行下面的操作:
把一个桶中所有的水倒掉;

把一个桶 A 中的水倒入桶 B，直到桶 A 空了或者桶 B 满了；

假设一开始容量为 3 升和 5 升的桶是满的，7 升的桶是空的，希望通过一系列操作使 3 个桶中任意一个中正好有 4 升水，那么至少需要[]次操作。

A.3 B.5 C.7 D.不可能

2. 程序设计与算法

2.1 实现如下编码算法，对于重复 2-9 次数的字符，用两个数字表示，即 NX（其中 N 为重复的次数，X 为重复的字符，下同），超过九个则先输出 9X，然后处理剩下的字符。对于连续的不重复的字符，则两边加 1 来封字符串。如果被封的字符串其中有数字为 1，则用 1 来转义。

示例：AAAAAABCCCC -> 6A1B14C, 12344 -> 11123124。。。 (下面的框架是用 C++ 语言写的。你可以用你熟悉的语言。)

```
void encode (const char* text, char* dest)
```

text 为需要编码的字符串，dest 表示编码输出的目标空间，而空间足够大

2.2 给定一颗有 n 个结点的二叉树。求它的所有结点数为 m 的连通子图数目。m<=n 分析你的算法的时间复杂度，解释算法即可，不必写代码。

用：假设有这样一种字符串，它们的长度不大于 26，而且若一个这样的字符串其长度为 m，则这个字符串必定由 a, b, c ... z 中的前 m 个字母构成，同时我们保证每个字母出现且仅出现一次。比方说某个字符串长度为 5，那么它一定是由 a, b, c, d, e 这 5 个字母构成，不会多一个也不会少一个。嗯嗯，这样一来，一旦长度确定，这个字符串中有哪些字母也就确定了，唯一的区别就是这些字母的前后顺序而已。

现在用一个由大写字母 A 和 B 构成的序列来描述这类字符串里各个字母的前后顺序：

如果字母 b 在字母 a 的后面，那么序列的第一个字母就是 A (After)，否则序列的第一个字母就是 B (Before)；

如果字母 c 在字母 b 的后面，那么序列的第二个字母就是 A，否则就是 B；

如果字母 d 在字母 c 的后面，那么……不用多说了吧？直到这个字符串的结束。

这规则甚是简单，不过有个问题就是同一个 AB 序列，可能有多个字符串都与之相符，比方说序列“ABA”，就有“acdb”、“cadb”等等好几种可能性。说的专业一点，这一个序列实际上对应了一个字符串集合。那么现在问题来了：给你一个这样的 AB 序列，问你究竟有多少个不同的字符串能够与之相符？或者说这个序列对应的字符串集合有多大？注意，只要求个数，不要求枚举所有的字符串。

嘿嘿，这就是你要解决的问题了。如果不嫌慢的话大可以穷举，不过这种解法拿出来那是显然不好意思和人打招呼的。事实上，如果设 AB 序列的长度为 n，那么这个问题是可以做到 $O(n^3)$ 的。

1、有两根不均匀分布的香，香烧完的时间是一个小时，你能用什么方法来确定一段 15 分钟的时间？

答：2 根香同时点燃，第一根两头都点燃，第二根只点一头，第一根点完的时候是半个小时，接着把第二根两头都点燃，第二根点完的时候就是 15 分钟。

2、一个经理有三个女儿，三个女儿的年龄加起来等于 13，三个女儿的年龄乘起来等于经理自己的年龄，有一个下属已知道经理的年龄，但仍不能确定经理三个女儿的年龄，这时经理说只有一个女儿的头发是黑的，然后这个下属就知道了经理三个女儿的年龄。请问三个女儿的年龄分别是多少？为什么？

答：2, 2, 9, 1 岁不可能

3、有三个人去住旅馆，住三间房，每一间房\$10 元，于是他们一共付给老板\$30，第二天，老板觉得三间房只需要\$25 元就够了于是叫小弟退回\$5 给三位客人，谁知小弟贪心,只退回每人\$1，自己偷偷拿了\$2，这样一来便等于那三位客人每人各花了九元，于是三个人一共花了\$27，再加上小弟独吞了不\$2，总共是\$29。可是当初他们三个人一共付出\$30 那么还有\$1 呢？

答：没错，三个人付了 27 块，老板拿了 25 块，小弟拿了 2 块

4、有两位盲人，他们都各自买了两对黑袜和两对白袜，八对袜子的布质、大小完全相同，而每对袜子都有一张商标纸连着。两位盲人不小心将八对袜子混在一起。他们每人怎样才能取回黑袜和白袜各两对呢？

答：不知道，还要仔细想想

5、有一辆火车以每小时 15 公里的速度离开洛杉矶直奔纽约，另一辆火车以每小时 20 公里的速度从纽约开往洛杉矶。如果有一只鸟，以 30 公里每小时的速度和两辆火车同时启动，从洛杉矶出发，碰到另一辆车后返回，依次在两辆火车来回飞行，直到两辆火车相遇，请问，这只小鸟飞行了多长距离？

答：记好两车相遇时间，就是鸟飞行时间，乘以其飞行速度就得到飞行距离。

6、你有两个罐子，50 个红色弹球，50 个蓝色弹球，随机选出一个罐子，随机选取出一个弹球放入罐子，怎么给红色弹球最大的选中机会？在你的计划中，得到红球的准确几率是多少？

答：不知道，还要仔细想想

7、你有四个装药丸的罐子，每个药丸都有一定的重量，被污染的药丸是没被污染的重量+1.只称量一次，如何判断哪个罐子的药被污染了？

答：不知道，还要仔细想想

8、你有一桶果冻，其中有黄色，绿色，红色三种，闭上眼睛，抓取两个同种颜色的果冻。抓取多少个就可以确定你肯定有两个同一颜色的果冻？

答：4

9、对一批编号为 1~100，全部开关朝上(开)的灯进行以下*作：凡是 1 的倍数反方向拨一次开关；2 的倍数反方向又拨一次开关；3 的倍数反方向又拨一次开关……问：最后为关闭状态的灯的编号。

答：不知道，还要仔细想想

10、想象你在镜子前，请问，为什么镜子中的影像可以颠倒左右，却不能颠倒上下？

答：人的眼睛是左右对称的

11、一群人开舞会，每人头上都戴着一顶帽子。帽子只有黑白两种，黑的至少有一顶。每个人都能看到其它人帽子的颜色，却看不到自己的。主持人先让大家看看别人头上戴的是什么帽子，然后关灯，如果有人认为自己戴的是黑帽子，就打自己一个耳光。第一次关灯，没有声音。于是再开灯，大家再看一遍，关灯时仍然鸦雀无声。一直到第三次关灯，才有劈劈啪啪打耳光的声音响起。问有多少人戴着黑帽子？

答：3

一、单选

1、80x86 中，十进制数-3 用 16 位二进制数表示为？

2、假定符号-、*、\$分别代表减法、乘法和指数运算，且

1)三个运算符优先级顺序是：-最高，*其次，\$最低；

2)运算符运算时为左结合。请计算 $3-2*4\$1*2\3 的值：

(A)4096, (B)-61, (C)64, (D)-80, (E)512

3、下列伪代码中，参数是引用传递，结果是？

```
calc(double p, double q, double r){q=q-1.0;r=r+p}
```

```
main(){
```

```
    double a = 2.5, b = 9.0;
```

```
    calc(b-a, a, a);
```

```
    print(a);
```

```
}
```

(A)1.5 (B)2.5 (C)10.5 (D)8 (E)6.5

4、求输出结果：

```
int foo(int x, int y){
```

```

    if(x <=0 || y <= 0) return 1;
    return 3 * foo(x - 1, y / 2);
}
printf("%d\n", foo(3, 5));
(A)81 (B)27 (C)9 (D)3 (E)1

```

5、下列哪个数据结构在优先队列中被最广泛使用？

(A)堆 (B)数组 (C)双向链表 (D)图 (E)向量

6、以下算法描述了一个在 n 个元素的双向链表中找到第 k 个元素的方法 ($k \geq 1$ 且 $k \leq n$):

如果 $k \leq n - k$, 从链表开始往前进 $k-1$ 个元素。

否则, 从终点出发, 往回走 $n - k$ 个元素。

这个算法的时间代价是？

(A) $\theta(n \log n)$ (B) $\theta(\max\{k, n - k\})$ (C) $\theta(k + (n - k))$

(D) $\theta(\max\{k, k - n\})$ (E) $\theta(\min\{k, n - k\})$

7、有一个由 10 个顶点组成的图, 每个顶点有 6 个度, 那么这个图有几条边？

(A)60 (B)30 (C)20 (D)80 (E)90

8、正则表达式 $L = x^*(x|yx)^+$ 。下列哪个字符串不符合 L

(A)x (B)xyxyx (C)xyx (D)yxx (E)yx

9、为读取一块数据而准备磁盘驱动器的总时间包括

(A)等待时间 (B)寻道时间 (C)传输时间 (D)等待时间加寻道时间

(E)等待时间加寻道时间加传输时间

二、算法

1、打印出一个二叉树的内容。

2、在一个字符串中找到第一个只出现一次的字符。如 abaccdeff, 输出 b。

3、给定一个长度为 N 的整数数组 (元素有正有负), 求所有元素之和, 最大的一个子数组。分析算法时空复杂度。不必写代码。

附上动态规划做法的答案:

最大子序列

问题:

给定一整数序列 A_1, A_2, \dots, A_n (可能有负数), 求 $A_1 \sim A_n$ 的一个子序列 $A_i \sim A_j$, 使得 A_i 到 A_j 的和最大

例如: 整数序列 -2, 11, -4, 13, -5, 2, -5, -3, 12, -9 的最大子序列的和为 21。对于这个问题, 最简单也是最容易想到的那就是穷举所有子序列的方法。利用三重循环, 依次求出所有子序列的和然后取最大的那个。当然算法复杂度会达到 $O(n^3)$ 。显然这种方法不是最优的, 下面给出一个算法复杂度为 $O(n)$ 的线性算法实现, 算法的来源于 *Programming Pearls* 一书。在给出线性算法之前, 先来看一个对穷举算法进行优化的算法, 它的算法复杂度为 $O(n^2)$ 。其实这个算法只是对穷举算法稍微做了一些修改: 其实子序列的和我们并不需要每次都重新计算一遍。假设 $\text{Sum}(i, j)$ 是 $A_i \dots A_j$ 的和, 那么 $\text{Sum}(i, j+1) = \text{Sum}(i, j) + A_{j+1}$ 。利用这一个递推, 我们就可以得到下面这个算法:

```
int max_sub(int a[],int size)
```

```
{
    int i,j,v,max=a[0];
```

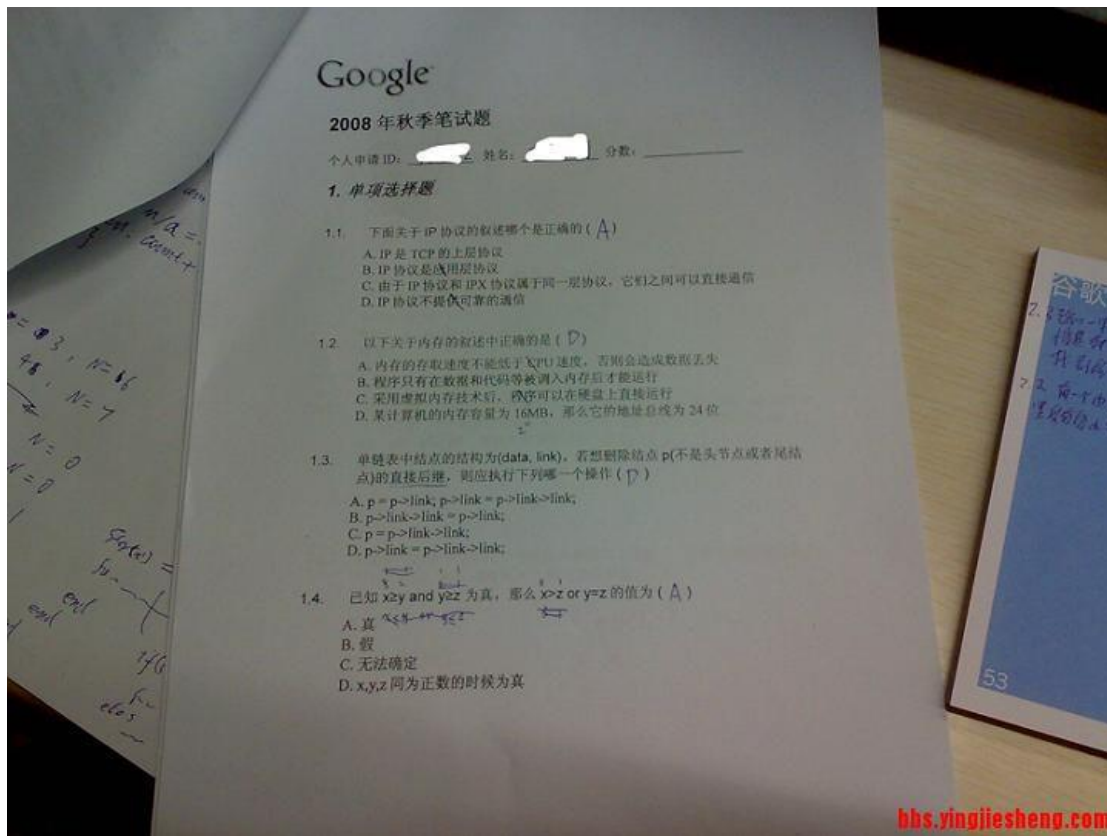
```
for(i=0;i<size;i++)
{
    v=0;
    for(j=i;j<size;j++)
    {
        v=v+a[j];//Sum(i, j+1) = Sum(i, j) + A[j+1]
        if(v>max)
            max=v;
    }
}
return max;
}
```

那怎样才能达到线性复杂度呢？这里运用动态规划的思想。先看一下源代码实现：

```
int max_sub2(int a[], int size)
{
    int i,max=0,temp_sum=0;
    for(i=0;i<size;i++)
    {
        temp_sum+=a;
        if(temp_sum>max)
            max=temp_sum;
        else if(temp_sum<0)
            temp_sum=0;
    }
    return max;
}
```

在这一遍扫描数组当中，从左到右记录当前子序列的和 `temp_sum`，若这个和不断增加，那么最大子序列的和 `max` 也不断增加(不断更新 `max`)。如果往前扫描中遇到负数，那么当前子序列的和将会减小。此时 `temp_sum` 将会小于 `max`，当然 `max` 也就不更新。如果 `temp_sum` 降到 0 时，说明前面已经扫描的那一段就可以抛弃了，这时将 `temp_sum` 置为 0。然后，`temp_sum` 将从后面开始将这个子段进行分析，若有比当前 `max` 大的子段，继续更新 `max`。这样一趟扫描结果也就出来了。

2.12 2009 年 google 笔试题选择题部分



Google

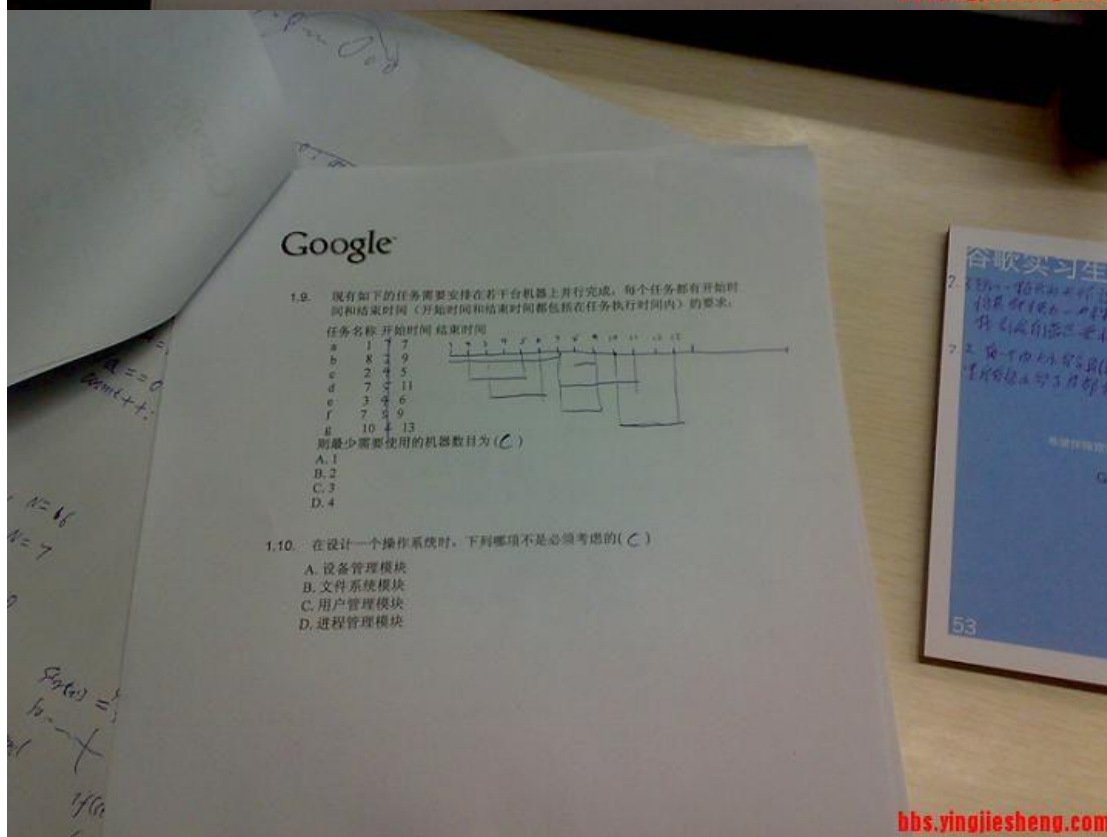
2008 年秋季笔试题

个人申请 ID: _____ 姓名: _____ 分数: _____

1. 单项选择题

- 1.1 下面关于 IP 协议的叙述哪个是正确的 (A)
- A. IP 是 TCP 的上层协议
 - B. IP 协议是成层层协议
 - C. 由于 IP 协议和 IPX 协议属于同一层协议, 它们之间可以直接通信
 - D. IP 协议不提供可靠的通信
- 1.2 以下关于内存的叙述中正确的是 (D)
- A. 内存的存取速度不能低于 CPU 速度, 否则会造成数据丢失
 - B. 程序只有在数据和代码等调入内存后才能运行
 - C. 采用虚拟内存技术后, 程序可以在硬盘上直接运行
 - D. 某计算机的内存容量为 16MB, 那么它的地址总线为 24 位
- 1.3 单链表中结点的结构为 (data, link), 若想删除结点 p (不是头结点或者尾结点的) 的直接后继, 则应执行下列哪一个操作 (P)
- A. $p \rightarrow link; p \rightarrow link = p \rightarrow link \rightarrow link;$
 - B. $p \rightarrow link \rightarrow link = p \rightarrow link;$
 - C. $p = p \rightarrow link \rightarrow link;$
 - D. $p \rightarrow link = p \rightarrow link \rightarrow link;$
- 1.4 已知 $x \geq y$ and $y \geq z$ 为真, 那么 $x \geq z$ or $y = z$ 的值为 (A)
- A. 真
 - B. 假
 - C. 无法确定
 - D. x, y, z 同为正数的时候为真

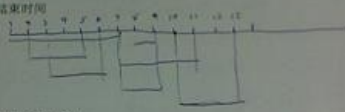
bbs.yingjiesheng.com



Google

1.9. 现有如下的任务需要安排在若干台机器上并行完成, 每个任务都有开始时间和结束时间 (开始时间和结束时间都包括在任务执行时间内) 的要求:

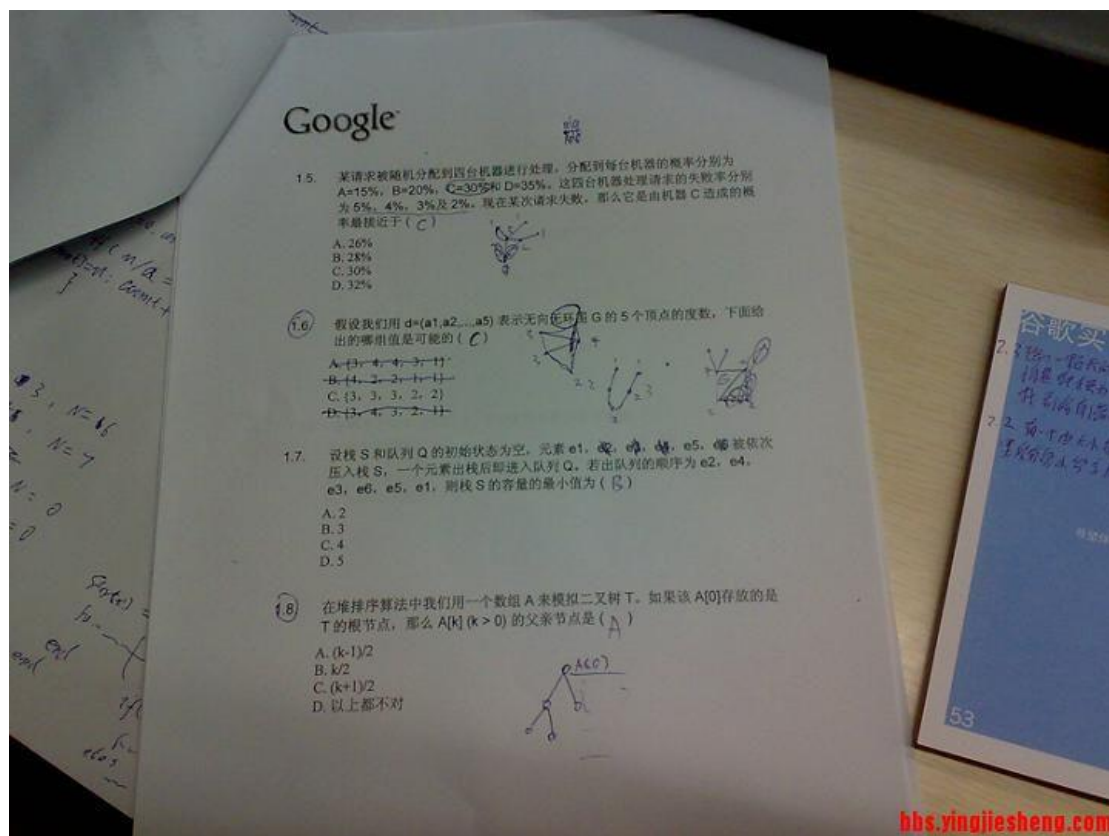
任务名称	开始时间	结束时间
a	1	7
b	8	9
c	2	5
d	7	11
e	3	6
f	7	9
g	10	13



最少需要使用的机器数目为 (C)

- A. 1
 - B. 2
 - C. 3
 - D. 4
- 1.10. 在设计一个操作系统时, 下列哪项不是必须考虑的 (C)
- A. 设备管理模块
 - B. 文件系统模块
 - C. 用户管理模块
 - D. 进程管理模块

bbs.yingjiesheng.com



第三章 Google 面试经验分享

3.1[面试] 软件开发工程师全英语面试

本文原发于应届生 BBS，发布时间：2016 年 7 月 20 日

地址：<http://bbs.yingjiesheng.com/thread-2048308-1-1.html>

面试一共四轮，全是 CODING。

第一轮是个香港帅哥，人很好，这轮是我表现最好的一轮。题目如下：

1.1. Tokenize a string to words. Ignore any space and punctuator

1.2. Design an distributed file system to store files of TB size

Follow-up: How to find and store the top-k most frequent keywords among documents stored on all Google servers

第二轮是他就问了很简单的一道题，然后就是不停地问我如何改进。

2. Given a list of words, find two strings S & T such that:

a. S & T have no common character

b. S.length() * T.length() is maximized

Follow up: how to optimize and speed up your algorithm

3.2[面试] google 16 Summer Intern

本文原发于应届生 BBS，发布时间：2016 年 7 月 20 日

地址：<http://bbs.yingjiesheng.com/thread-2048307-1-1.html>

一共两轮面试，第一轮问了一道算法题，内容跟 leetcode 一个原题很类似，是 read4 byte 的变体。第二轮问了两个 easy 题，主要是 string manipulation。总体来说比较简单，问了一下时间复杂度和空间复杂度的问题。

3.3[面试] 软件工程师面试

本文原发于应届生 BBS，发布时间：2016 年 7 月 20 日

地址：<http://bbs.yingjiesheng.com/thread-2048305-1-1.html>

第一轮电面，老外，美国口音：

1，很多文件里面有很多数字，设计排序的算法，答了个外排序的归并，接着问很多机器怎么进一步优化。楼主不懂 Map-Reduce 就 YY 了一通。那边又要求希望两个 Iterator 做完所有排序，然后瞎扯了一会这题就这么过了。。。
2，写一个函数 Children(Node *r) 输出 BST 的当前节点的所有儿子。递归和非递归，哪个快怎样优化之类的问了问，就结束了。

第二轮电面，中文：

1，FindCloset(float a[],length,target)，有序表找最接近数字，这个简单二分查找，写完他也没说什么。

2，数组加一个数，比如[2,3,4,5] + 45=[2,3,9,0]。DT 的是只能用数组不能用 vector，在加完还有进位的时候需要重新 new 空间，很快写完以后又让优化了几遍。onsite 两轮后跪了，各一个算法题，都是中文，估计跪在了第二轮上，

1，给一个二叉树，让找出所有相同的子树。

先说了枚举所有节点对然后递归判断的 n^3 简单方法，面试官不满意，然后用 memo 优化到 $O(n^2)$ ，写完解释了一会就算结束了。

2，一些人排成队，每个人知道自己前面有多少个人比自己高。已知每个人的身高。要求根据这些信息求出原先排好的队。

贪心算法，证明了挺久，最后写完解释了一下就说时间到了，估计想的太久了就被 BS 了。

3.4[面试] 记一次 Google 面试经历

本文原发于应届生 BBS，发布时间：2016 年 6 月 30 日

地址：<http://bbs.yingjiesheng.com/thread-2106634-1-1.html>

这是我上周去面试的地方。很顺利，我觉得——至少我认为我已经尽我所能，并且无论发生什么事情对我都是有帮助的。

由于我签署了保密协议，所以我不能把他们问我的问题写出来。但是，我可以写我大概的面试过程，以及我做了哪些准备。

首先，这是谷歌——所以我当然不会让我在那里工作的朋友不推荐我。我没有经过电话采访，因为我是“本地”的，所以我被直接邀请到公司面试。我个人认为，面对面绝对比电话面试好，我很幸运。

首先，我和招聘人员很简短地聊了一下，他告诉我我需要使用白板和进程，然后是两个软件工程师分别对我进行了 45 分钟一对一的两场面试。然后我很惊讶自己居然发现了面试的乐趣！通常情况下我在面试时感受到的只有压力，但是 Google 的面试真的很棒，因为没有笨蛋的参与，不会有对牛弹琴的尴尬。超酷！第一场面试的问题相当容易，而第二场则有点难，但也不算太糟糕。可能最困难的是在白板上（而非计算机！）上写代码——在白板上很难做 TDD！他们告诉你没有必要盛装出席，因此绝对不要装扮仪表——因为你需要坐在地板上写代码，而且不是一时半会就可以在白板上写完的，还会写得满手都是油墨。想象一下如果你穿着裙子会怎么样！

他们发给我了一系列很有帮助的面试准备。由于我坚实的学习基础，所以在一定程度上我都有所涉猎。此外平时我还会做这些事情——及时了解最新的行业新闻，广泛使用 Google 产品，写博客，以及思考技术如何改变我们的生活（是的，最后两件事情很有帮助）。我会深刻理解和使用的 Java 5 的新产品，例如 Generics, Enums, for-each 等。

下面是我准备的内容（所有的图书链接均链接到 Amazon）：

阅读《Effective Java》（第 2 版）——不带任何夸大之词，这本书助我成为了一名合格的 Java 程序员。

阅读《Programming Interviews Exposed（编程面试攻略）》，并通过所有练习——概述和复习基本的数据结构，如列表和树。递归部分没有我自己学到的那么强大（因为我有函数背景），它反而主张编写迭代方法，不过除此之外，这的确是一本既非常好又有用的书。

阅读《Coders at Work》——这本书中谈到了很多的谷歌人，表达了作者对文化的观点。它还介绍了很多我不知道的却又非常有趣的编程历史，以及关于这些历史伟人如何解决问题的见解，以及对于 API、可扩展性等的讨论。

《Combinatorial Algorithms: Generation, Enumeration, and Search》（我几乎看过整本书，并上过一门关于组合算法（Combinatorial Algorithms）的课程）——说实话，我不是这本书的狂热爱好者。我觉得这本书的数学符号气息太重。因为我们是程序员，而不是数学家，所以使用实际代码来解决问题可能更有帮助，并且维基百科几乎可以肯定更具可读性。

研究《Java Puzzlers》——有助于培养钻研代码，研究问题的心态。虽然并没有人要求我这么去做，但我确实需要批判性地看待自己的代码。IBM 的面试中就提到了关于 EB 手机屏幕的问题，而且谷歌也用了那一类的问题。

复习并发问题——死锁、活锁、互斥锁、内核锁、信号量等。什么时候在 Java 中使用 synchronized 关键字？如何避免死锁？如何避免活锁？

复习树的遍历——前序，中序，后序。深度优先搜索 vs. 广度优先搜索。A*，Dijkstra 算法等。

复习平衡二叉树——红黑树、AVL 树、伸展树。

复习图表。表示图，最小生成树，搜索等。

运行时分析。

6 种排序算法编码——包括关键的时间复杂度为 $n \log n$ 的算法——TDD 风格（测试驱动开发——关于我的测试案例请看这个帖子）。

哈希表编码，只使用数组。包括：泛型，动态数组，延迟初始化。此外还有测试先行。

做一做所有手头可以获取的实践问题——搜索“谷歌面试问题”，但不要浪费时间在什么面试预测或井盖问题上，要找类似问题——有时，我在 Eclipse 中编码，但有时在 Google 文档中。我喜欢和朋友一起工作，他会审查我的代码，并提出问题。

和已经在那里工作多年的朋友交谈。问很多问题。他真的很赞，帮我做了很多的准备工作。不仅如此，了解他为什么认为我会是一个不错的求职者，以及他为什么会相信我可以做好，有助于我知道自己为什么要在那里工作（这是一个老生常谈的问题，虽然这家公司是 Google，但正如我的一个导师说的那样，你想为他们工作的热切程度得

和他们想要你的程度差不多)。

我的 Google 朋友说我做了“疯狂的准备”，甚至可以说我做了充分的准备——那么，除此之外，我还做了什么呢？

更多地运行时分析——尽可能多找到代码进行分析。

计算总和。例如，如何计算 1 至 n 的和？事实证明是有用的。第二次面试中就涉及到了回放运行时分析，因为我有一些东西看起来是这样的： $(n-1)(n-2) + (n-2)(n-3) + \dots + (3)(2) + (2)(1)$ 。当然，我并没有在当时创建它，所以它的时间复杂度上限为 $O(n^2)$ 。

复习 Java 库的数据结构。有一回，我说了这样的话，“我知道一个不重复的数据结构，但我现在一时想不起它的名字”。后来在当天下午我想起来了。

复习库方法，复习一些关键的东西——数组和字符串将会很有帮助。

在白板或纸上练习编码。也许你理所当然地认为插入行或重构很简单，事实上，当你在白板上写代码的时候，肯定没有在计算机上趁手。而且我们很容易忘记返回语句——Eclipse 中不需要我们这么做。因此我倾向于先写声明和返回语句，然后再在中间补上代码——当然，在白板上可不能这么做！

接下来要说的是？

Waiting。

Waiting。

Waiting！

无论发生什么，我都希望能得到一些反馈。

编码面试问题（以及测试用例！）

完成《Java Puzzlers》。

充分利用时间——我花了很多时间来做好诸多准备工作。

调查其他公司。特别是——与 IBM 员工见面，看看有没有什么东西适合用于 Google。

3.5 [面试] 字符串转置匹配 - 面试 Google 工程师必问的 15 道面试题

本文原发于应届生 BBS，发布时间：2016 年 6 月 30 日

地址：<http://bbs.yingjiesheng.com/thread-2106649-1-1.html>

1. 请写出一个整数乘法的算法

挑战：这是一个相当开放性的问题，设计初衷是看看工程师是否会定义参数。编程语言是什么？是汇编语言吗？是：那么你得白手起家开发出一套基本运算来。

否则的话：那就简单了，只需将数字套进去即可，因为大部分语言均支持数学运算。

2. 合并两个排序链表

挑战：

这类问题是 Google 面试问题的一个共同趋势：找出解决问题的有效办法。合并两条链表是一般会在链表之间发

生“冲突”（因为它们各自有特定的次序，而你的合并会把次序搞乱）你必须找出一种算法快速消除那些冲突。

3.给出一组区间（以秒计），让你找出重叠的部分

挑战：**Google** 一般也会要你解决一堆数学问题。这些设计的意图是让工程师对问题进行有效的思考，而不是仅仅知道如何对每个问题进行编码。此处的问题会要你实现一个聪明的数学技巧。

4.如果你有 n 个节点的列表，对于一个有向无环图来说，最多可以有几条边？

挑战：这是著名的数学问题旅行商（**traveling salesman problem**）问题的一个派生。这是一个不要求你找出聪明的数学算法而是要你拓展思路的数学问题。你必须在节点间画线，条件只有一个：无论你从哪里开始，永远不要回到原点

5.在 Java 里面，**finally**、**final** 和 **finalize** 的区别是什么？

挑战：看看你知不知道这些术语。比方说，**Finalize** 处理的是“垃圾回收”，而 **Finally** 则是指向一个特定的错误。

6.从一大块文本中去掉重复的行

挑战：一个基本问题，看你是不是能估算出问题是否可以在有限的时间内解决。这样的问题通常可以采取暴力攻击手段，但却有可能永远都无法解决。

7.给你一串字符，要你找出包含有给定字符集的最小窗口

挑战：此类问题有许多解决方式，这完全取决于你选择的算法是什么。**Google** 要找的是能选出更快算法的工程师。

8.写一段程序计算出某字符串（/矩阵）是否另一字符串（/矩阵）的旋转（/转置）

挑战：这个问题靠的是看你是不是能够灵活掌握一些计算技术的转义。比方说，“**strings**”一语通常是指由字符、单词或可打印、可见的东西组成的多行字符串。不过同时也可以是指矩阵（二维对象）及别的对象。你必须检查看看它是不是可以旋转（转置），然后跟另一个进行对比。

9.旧金山能容纳多少个气球？

挑战：此类问题 **Google** 的面试官一般都会问到，目的是想看看你设定的“边界条件”是什么样的。这个问题有许多解决方式，可以通过添加限定条件来进行简化—比如说旧金山的面积，这种限定在进行编程的时候是很重要的。

10.粘滞位是什么？为什么要用？

挑战：粘滞位是一个跟设置和权限有关的系统管理员术语。**Google** 想测试一下，看看你是否了解一些管理的细节，而不是只会编程。

11.给出一组整数，值范围在很窄的区间，但是区间未明，如何找出其中位数？

挑战：又是一个要你实现聪明的数学技巧的问题，离散数学课你有没有认真听？这可不是什么简单问题。

12. 给出一组区间，找出交集数最多的一个

挑战：现在 Google 要你确信知道自己检查的是问题的正确部分，无视没有价值的部分，这样才能够令算法更高效。13. 100 个整数组成的数组，次序被打乱。其中有一个数被拿掉了，把它找出来

13. 100 个整数组成的数组，次序被打乱。其中有一个数被拿掉了，把它找出来

挑战：运用另外一个数学技巧可以几行代码搞定。测试的目的是看看你是不是不仅知道用什么办法，还知道如何实现代码。

14. 判断某个数独解决方案是否正确

挑战：这个要你想出办法用另一种数学技巧从算术上快速检验解决方案。

15. 给你一支股票的日价，找出一一次买卖交易 (buy-sell trading) 如何操作才会损失最大

挑战：这个问题非常棒，你必须竭尽所能，把所有的知识本领都用上。看看边界条件（即挑出变动最大的股票价格点）然后有效地算出波动最大的一次。我们能想到的唯一像样一点的办法是暴力计算，把每一个股价都遍历一遍。

3.6 [面试] Google 两次面试的经验 (phone+onsite)

本文原发于应届生 BBS，发布时间：2016 年 6 月 30 日

地址：<http://bbs.yingjiesheng.com/thread-2106655-1-1.html>

去年今年面了两次 Google，特来提供两次的面试和一点感想，lc # 是 leetcode 题号的意思： 2015 年：

phone: lc 329

onsite:

1. 设计 battleship 这个游戏，似乎是个经典游戏，可我完全没玩过。。

2. 设计贪吃蛇游戏

3. lc 42 和 lc 200

4. lc 295

最后跪了，HR 告诉我是第一轮 battleship 这题 feed back 不好，其他轮都很好，第一轮面试官说我没有选择最佳的数据结构。

2015 年那时是刚毕业，刷了几遍题，没怎么准备设计题，结果 onsite 迎面两轮设计，算法题倒是都很容易。。

然后工作了一年左右到了 2016 年，又去面 Google: .1point3acres 组

phone: lc 354 . From 1point 3acres bbs

onsite:

1. Implement Caesar cypher，很容易，面试官像个教授，讲了很多密码学历史（我并没有相关背景）不像在面试，

像听 lecture...

2. lc 140

然后是一道关于按照固定的顺序重新排序，多少次排序才能回到初始顺序的题目，我们总是把第 x 个位置的元素移到第 y 个位置，比如【0, 1, 2, 3】就是完全不换顺序，一次排序就回去了；【3, 2, 1, 0】是位置 3 和位置 0 互换，位置 2 和位置 1 互换，两次排序就能回到原顺序；而【1, 2, 3, 0】就是每次把一个位置上的元素往后移动一个位置，那显然转一圈就回来了。

3. 好几道小题，都非常简单，比如有 lc 345 然后让我设计 Youtube。。因为时间剩下不到一半，所以是比较粗略的设计。

4. 超级老的老爷爷，问过去的 project 和经历，问一些基本数据结构的概念，比较不同结构、算法的优劣，二叉树和图论比较多，问题很基础，比如 DFS 和 BFS 的比较。哦，还有 merge sort 和 quick sort。

5. 老奶奶，系统设计题，扯了好久的 cache 和一点点 load balancer 的设计。

我觉得整个面试过程中没有太难的算法题，就算碰到一道也只是整个面试里很小一部分，去年面的时候已经感到算法题不多了，今年面感觉更少，所以可能多加强沟通能力，扩展知识面比较重要吧。

另外我去年问了一个面试官，从哪来的题目，Google 有没有题库，他说他就是上网搜搜，看最近有哪些有趣的题目就出自己感兴趣的那个。。

去年题目刷了不少感觉也没啥用，于是我今年面 Google 前就没怎么刷题，看了些技术博客，不过也就是随便看看，被问到时还是主要靠瞎扯。。当然，看过相关的文章被问时还是会好一点，今年这次的面试结果还没出来，太多的 open question，感觉不出自己扯的好不好，希望会有好结果吧。。

补充内容 (2016-6-17 09:06):

对了，楼主两次面试都不是在 MTV，分别是 Venice 和 San Bruno，可能 MTV 的算法题会比较多？

3.7 陈广琛：面试体验：Google 篇

本文原发于应届生 BBS，发布时间：2012 年 8 月 10 日

地址：<http://bbs.yingjiesheng.com/thread-1474981-1-1.html>

尝试在自己的博客上搜索点东西，结果发现 4 年多以前还在博客上写过一系列的 recruiting events，把大四时候参加过的各种笔试面试都记录下来。我从去年准备离开百度开始，到现在总过面试过 4 家公司：Google、Microsoft、Yahoo、Facebook，原本去年也想把面试经验写一写的，结果一拖就拖到现在。我不想写面试经验，因为我个人不喜欢漏题和背题的做法。我自己作为面试官，知道要设计出来一道好用的题目有多难，所以我希望面试者都是如实表现自己解题能力的。我更喜欢写面试体验，就是在整个面试过程中一家公司给人的印象是怎样的，HR 和面试官是否专业，能否让人信服这是一家值得长期工作的公司。

我想写的第一家公司是 Google，因为它是我在想要离开百度时第一家联系到我的公司。2010 年 12 月底的某一天早上，我突然感觉到我应该离开百度，因为如果这个时候已经没有勇气离开这家公司了，很可能就不会再想要离开了。当天中午在百度大厦西餐厅吃午饭，接到一个 Google 上海 HR 的电话，问我有没有兴趣去面试，我想既然你打电话来的时机那么好，我就答应你去面试吧。（在那一天之前，我对猎头的标准回复是「有美国或者香港的职位吗？」）她问我将来希望在北京还是上海工作，当时我对北京的厌恶程度还没有现在那么高，同时觉得搬家到上海又比较麻烦，于是就说在北京，接着我就变成跟北京 HR 沟通了。

Google 的 HR 会负责做两件简单得不需要面试官做的事情，这能够很好的提高招聘流程的效率。第一件是确认你能够适应工作环境中的英语，为此 HR 要我用英语跟她对话两三分钟，主要就是让我说说工作经验和其中的亮点。习惯

在私企工作的人不要以为外企对英语的要求很高，其实大多数长期在中国工作的人说话或者发邮件都会很 Chinglish 啦，所以关键是要敢于用英语进行沟通。

然后 HR 发了一个 Codility 的地址给我，让我有空抽时间去做题。一个小时 3 道难度相当于 OI 基础题的题目，平均 20 分钟一道。最简单的题目一看就知道是 $O(n)$ 能解决的，最复杂的题目看上去是 $O(n^2)$ 但想一下就能优化为 $O(n \log n)$ 。对于有算法训练背景的人来说，这样的题目会让人感觉到很有把握。对于没有经受过算法训练的人来说，掉进陷阱里是很容易的。很可能没有把 $O(n^2)$ 优化为 $O(n \log n)$ ，结果超时；可能没仔细看题目说明的数值取值范围，某些变量选错了数值类型，结果溢出。考虑到 Google 重视算法的程度，再加上 Google 中国面试的额外难度，算法训练还是很必要的。

在我通过 Codility 测试后，HR 问我对题目难度的反馈，然后约了一轮电话面试，并且告知面试主要围绕算法、数据结构、系统设计、编码来进行。Google 面试的格式都很固定，45 分钟内期望你能做出 3 道题来。这 3 道题最起码要能把人人都能想出来的「笨办法」用代码写出来，否则会让面试官感到不满意。如果有些题目能够比较快地做出来，面试官就会让你优化。就算你第一次给出的答案已经是业界已知最优解，面试官都还是会让你优化，因为谁也不知道有没有人能在面试过程中突然爆发，想出一些过去没人想到过的解法。如果面试官心中已有优化的方案，在你想不出优化方案时他可能会给你提供一些提示。

一轮电话面试后，HR 就开始约到 Google 办公室的面试了。第一次约了下午 3 轮面试，还是那个很固定的格式：每轮面试 45 分钟，两轮间隔 15 分钟。整个面试流程让人感觉到很人性化：在 Google 签到后，HR 会先带你去 kitchen 拿点吃的喝的，然后把你带到面试所用的会议室。多轮面试的话，HR 中间还会来问一下你要不要去洗手间，或者多拿两瓶水。面试完毕后 HR 会来问你感觉如何，同时也会让你知道面试官的初步反馈是否跟你的感觉一致。我在 3 轮面试中有一轮感觉不太好，因为面试官只给了 2 道题，并且我最终都没办法解出来，HR 也确认了就是这一轮的反馈不好。

此外，Google 的招聘流程还让人感觉到很有效率。作为面试官，我也知道自己写面试反馈有多喜欢拖延，而且公司填写面试反馈的系统越不人性化我就越想要拖延，然而公司内部系统做得人性化的又实在罕见。Google 的面试基本上隔天就有结果，然后 HR 就会约下一轮的面试。因为我在百度的时候每周哪个时间没有会议是很确定的，所以我总是选择下周同一个时间段来面试。在经过总共 4 轮面试后，HR 说因为前面有一轮的面试官反馈不好，所以希望再加一轮面试。因为前面反馈不好的面试官比较 senior，所以这次找了一位同样 senior 的面试官来面试，于是我又去了一次 Google 办公室。

完成 5 轮面试后，HR 把材料提交给 Google 的北京招聘委员会，结果没有通过。HR 说，因为 Google 都是按照后端工程师的标准来招聘，看重算法和数据结构，前端工程师要通过不容易。因为 Google 没有专门的前端工程师，只有一个软件工程师职位，所以所有人还是必须按照一个标准来衡量。她问我如果找到专门需要前端工程师的团队，并且需要额外再面试的话，我是否感兴趣。当时 Google 是我的第一选择，我当然说感兴趣啦。

后来 HR 跟我说，她帮忙问过 Google Maps，可惜对方说不要专才只要通才。又过了几个星期，HR 发现 IME 需要专门做前端的人，于是帮我再约了一轮面试。这轮面试是在 Google 办公室做的，但实际上是视频会议，因为面试官在美国。（不确定面试官是在美国出差，还是美籍华人。）面试过程跟电话面试类似，用 Google Docs 写代码，比电话面试要好的是说话时能够见到人。

这一轮面试结束后，我的材料再次进入 Google 的北京招聘委员会。HR 说这次专门找了对前端有经验的人来审阅我的材料，结果顺利通过了。接着 HR 问我要了一大堆的补充材料，包括高考成绩和 GPA(连同成绩单)，还包括当前薪酬和竞争对手的 offer(我当时有 Yahoo 的 offer)，甚至包括过去的获奖和晋升经历。所有这些材料都会发往 Google 美国总部审阅，具体流程 HR 没有细说，但看 Don Dodge 的文章可以了解一些。最后我被 Google 美国总部给拒绝了，然后 HR 还是一如既往地及时沟通，并且安慰了我几句。

整个 Google 招聘流程下来，可以感觉到人性化和高效率，同时也能感觉到 HR 确实在很努力地地为候选人争取机会。可以说，无论是否通过，Google 招聘流程至少能给候选人一个很好的印象。据我所知，尽管 Google 声称全球招聘标准一致，但因为中国聪明且懂算法的人实在太多，所以难度更高是很正常的。能够在 Google 中国以外的地区应聘的话，应该会容易一些。

3.8 一次谷歌面试趣事

本文原发于应届生 BBS，发布时间：2011 年 4 月 11 日

地址：<http://bbs.yingjiesheng.com/thread-895281-1-1.html>

很多年前我进入硅谷人才市场，当时是想找一份高级工程师的职位。如果你有一段时间没有面试过，根据经验，有个非常有用的提醒你应该接受，就是：你往往会在前几次面试中的什么地方犯一些错误。简单而言就是，不要首先去你梦想的公司里面面试。面试中有多如牛毛的应该注意的问题，你可能全部忘记了，所以，先去几个不太重要的公司里面面试，它们会在这些方面对你起教育（再教育）作用。

我第一家面试的公司叫做 `gofish.com`，据我所知，`gofish` 这家公司如今的情况跟我当时面试时完全的不同。我几乎能打保票的说，当时我在那遇到 的那些人都已不再那工作了，这家公司的实际情况跟我们这个故事并不是很相关。但在其中的面试却是十分相关的。对我进行技术性面试的人是一个叫做 `Guy` 的 家 伙。

`Guy` 穿了一条皮裤子。众所周知，穿皮裤子的面试官通常是让人“格外”恐怖的。而 `Guy` 也没有任何让人失望的意思。他同样也是一个技术难题终结者。而且是一个穿皮裤子的技术难题终结者 —— 真的，我做不到他那样。我永远不会忘记他问我的一个问题。事实上，这个问题是非常的普通 —— 在当时也是硅谷里标准的面试题。问题是这样的：

假设这有一个各种字母组成的字符串，假设这还有另外一个字符串，而且这个字符串里的字母数相对少一些。从算法是讲，什么方法能最快的查出所有小字符串里的字母在大字符串里都有？

比如，如果是下面两个字符串：

String 1: ABCDEFGHLMNOPQRS

String 2: DCGSRQPOM

答案是 true，所有在 string2 里的字母 string1 也都有。如果是下面两个字符串：

String 1: ABCDEFGHLMNOPQRS

String 2: DCGSRQPOZ

答案是 false，因为第二个字符串里的 Z 字母不在第一个字符串里。

当他问题这个问题时，不夸张的说，我几乎要脱口而出。事实上，对这个问题我很有信心。（提示：我提供的答案对他来说显然是最糟糕的一种，从面试中他大量的各种细微表现中可以看出）。

对于这种操作一种幼稚的做法是轮询第二个字符串里的每个字母，看它是否在第一个字符串里。从算法上讲，这需要 $O(n*m)$ 次操作，其中 n 是 string1 的长度， m 是 string2 的长度。就拿上面的例子来说，最坏的情况下将会有 $16*8 = 128$ 次操作。

一个稍微好一点的方案是先对这两个字符串的字母进行排序，然后同时对两个字串依次轮询。两个字串的排序需要（常规情况） $O(m \log m) + O(n \log n)$ 次操作，之后的线性扫描需要 $O(m+n)$ 次操作。同样拿上面的字串做例子，将会需要 $16*4 + 8*3 = 88$ 加上对两个字串线性扫描的 $16 + 8 = 24$ 的操作。（随着字串长度的增长，你会发现这个算法的效果会越来越越好）

最终，我告诉了他一个最佳的算法，只需要 $O(n+m)$ 次操作。方法就是，对第一个字串进行轮询，把其中的每个字母都放入一个 Hashtable 里（成本是 $O(n)$ 或 16 次操作）。然后轮询第二个字串，在 Hashtable 里查询每个字母，看能否找到。如果找不到，说明没有匹配成功。这将消耗掉 8 次操作 —— 这样两项操作加起来一共只有 24 次。不错吧，比前面两种方案都要好。

`Guy` 没有被打动。他把他的皮裤子弄的沙沙响作为回应。”还有没有更好的？”他问道。

我的天？这个家伙究竟想要什么？我看看白板，然后转向他。”没有了， $O(n+m)$ 是你能得到的最好的结果了 —— 我是说，你至少要对每个字母至少访问一次才能完成这项操作 —— 而这个方案是刚好是对每个字母只访问一次“。我越想越确信我是对的。

他走到白板前，”如果这样呢 —— 假设我们有一个一定个数的字母组成字串 —— 我给每个字母分配一个素数，

从 2 开始，往后类推。这样 A 将会是 2，B 将会是 3，C 将会是 5，等等。现在我遍历第一个字符串，把每个字母代表的素数相乘。你最终会得到一个很大的整数，对吧？然后 —— 轮询第二个字符串，用每个字母除它。如果除的结果有余数，这说明有不匹配的字母。如果整个过程中没有余数，你应该知道它是第一个字符串恰好的子集了。这样不行吗？”

每当这个时候 —— 当某人的奇思异想超出了你的思维模式时，你真的需要一段时间来跟上他的思路。现在他站在那里，他的皮裤子并没有帮助我理解他。

现在我想告诉你 —— Guy 的方案（不消说，我并不认为 Guy 是第一个想出这招的人）在算法上并不能说就比我的好。而且在实际操作中，你很可能仍会使用我的方案，因为它更通用，无需跟麻烦的大型数字打交道。但从”巧妙水平“上讲，Guy 提供的是一种更、更、更有趣的方案。

我没有得到这份职位。也许是因为我拒绝了他们提供给我的一些讨厌的工作内容和其它一些东西，但这都无所谓了。我还有更大更好的目标呢。

接着，我应聘了 become.com。在跟 CTO 的电话面试中，他给我布置了一道”编程作业“。这个作业有点荒唐 —— 现在回想起来，大概用了我 3 天的时间去完成。我得到了面试，得到了那份工作 —— 但对于我来说，最大的收获是这道编程作业强迫我去钻研并有所获。我需要去开发一个网页爬虫，一个拼写检查/纠正器，还有一些其它的功能。不错的东西。然而，最终，我拒绝了这份工作。

终于，我来到了 Google 面试。我曾说过 Google 的面试过程跟外面宣传的很一致。冗长 —— 严格，但诚实的说，相当的公平。他们在各种面试过程中尽最大的努力去了解你、你的能力。并不是说他们在对你做科学研究，但我确信他们是努力这样做。

我在 Google 的第四场面试是一个女工程师，老实话，是一场很无聊的面试。在前面几场面试中我表现的很好，感觉到我的机会非常的大。我相信如果不做出什么荒唐事情来，十拿九稳我能得到这份工作。

她问了我一些关于排序或设计方面的非常简单的问题，我记不清了。但就在 45 分钟的面试快要结束时，她对我说”我还有一个问题。假设你有一个一定长度的由字母组成的字符串。你还有另外一个，短些。你如何才能知道所有的在较短的字符串里的字母在长字符串里也有？”

哇塞。Guy 附身了。

现在，我完全可以马上结束这场面试。我可以对她说“哈哈，几个星期前我就知道答案啦！”，这是事实。但就是在几个星期前被问到这个问题时 —— 我给出的也是正确的答案。这是我本来就知道答案的问题。看起来就好像是 Guy 为我的这次面试温习过功课一样。而且，可恶，人们通常是通过上网来搜集面试问题 —— 而我，我可以毫不客气的说，对于这些问题，我不需要任何“作弊”。我自己知道这些答案！

现在你们可能认为——就在她问出了问题之后，在我准备开始说出在脑海里构思完成的最后的演讲之前——你们可能会想，我应该是，当然该，从情理上讲，镇定的回答出这个问题，并且获得赞赏。可糟糕的是，事实并不是这样。打个比喻，就像是她问出来问题后，我在闹子里立即举起了手，并大叫着“我！嗨！嗨！我知道！让我来回答吧！”我的大脑试图夺走我对嘴巴的控制权（这事经常发生），幸亏我坚强的毅力让我镇定下来。

于是我开始回答。平静的。带着不可思议的沉着和优雅。带着一种故意表现出来的 —— 带着一种，我认为，只有那种完全的渊博到对古今中外、不分巨细的知识都精通的人才能表现出来的自信。

我轻描淡写的说出来那种很幼稚的方案，就好像是这种方案毫无价值。我提到了给它们排序，就好像是在给早期的《星际迷航》中的一个场景中的人物穿上红 T 恤似的。最后，平淡的，就好像是我决定了所有事情的好坏、算法上的效率，我说出了 $O(n+m)$ 一次性方案。

我要告诉你——尽管我表明上的平静——这整个过程我却在做激烈的挣扎，内心里我在对自己尖着——“你个笨蛋，赶紧告诉她素数方案！”

当我完成了对一次性算法的解释后，她完全不出意外的认可的点了下头，并开始在笔记本上记录。这个问题她以前也许问过了一百次，我想大部分的人都能回答上来。她也许写的是“回答正确。无聊的面试。正确的回答了无聊的字符串问题。没有惊喜。无聊的家伙，但可以留下。”

我等了一会。我让这种焦灼的状态持续的尽可能的长。我可以发誓的说，如果再耽搁一分钟，我一定会憋出脑血栓、脱口说出关于素数的未解之谜。

我打破了沉默。“你知道吗，还有另外一个，可能是更聪明的算法。”

她二目空空的抬头看了一眼，仅在瞬间闪过一丝希望。

“假设我们有一定长度的字符串。我们可以给每个字母分配一个素数，从 2 开始。然后我们把大字串中的每个字母代表的素数相乘得出一个数，用小字串中的每个字母代表的素数去除它。如果除的过程中没有产生余数，则小字串是大字串的一个子集。”

在此时，我猜，她看起来就像是 Guy 当时把相同的话说给我听时我表现出来的样子。而我演讲时泰然自若的表情没了，眼睛瞪大，说话时稍微带出来一些唾沫星子。

一会儿后，她不得不说了，“可是...等一下，有可能...是的，可以这样！可是如何...如果...噢，噢，可行！简洁！”我得意洋洋的吸了一口气。我在我的面试记录里写下了“她给了我一个‘简洁’的评语！”在她提出这个问题之前我就确信能得到这份工作，现在我更加确信了。还有一点我十分确信的是，我（更准确的说是 Guy）给了她今天的好心情。

我在 Google 干了 3 年，生活的十分愉快。我在 2008 年辞职去到一个小公司里做 CTO，之后又开办了一个自己的公司。大概是一年前，我偶然的在一个创业论坛上遇到了 Guy，他记不得我了，当我向他细述这段往事时，他对他那条皮裤子大笑不已。

话说回来，如果这个故事里有什么教育意义的话——永远不要冒失的首先去应聘你梦想的公司，应先去应聘那些你不看好的职位。你除了能从这些面试中获得经验外，你说不定能遇到某个能为你的更重要的面试铺路的人呢。事实上，这个经验在你生活中的很多其它事情上也适应。

说正经的，如果你有机会想找一个解决问题的高手——雇佣 Guy 比谁都强。那个家伙很厉害。

（在这些陈年旧账里发现的一点技术瑕疵：字母有可能重复而字符串可能会很长，所以必须要有统计。用那个最幼稚的解决方案时，当在大字符串里找到一个字符后就把它删掉，当这样仍然是 $O(n*m)$ 次。在 Hashtable 里我们会会有一个 key-》value 的计数。Guy 的方案在这种情况下仍然好用。）

修改：11/30/10 —— 本文中提到的 Guy 看到了这篇文章，并在评论中做了澄清。值得一读。

3.9 [面试] 软件工程师面试经验

本文原发于应届生 BBS，发布时间：2017 年 6 月 30 日

地址：<http://bbs.yingjiesheng.com/thread-2106642-1-1.html>

早上去公司有 hr 接待，然后有 4 轮面试，中午有饭吃。面试主要面试算法，同时会问一些系统设计。有 1 面是英文面试。整个工程很轻松，发挥出自己就好。

主要是一些算法和系统设计如：

怎么计算两个字符串公共的最大长度。如何设计一个微博的后端系统，如何设计一个微信，如何做一个电商的后端

3.10 google 应聘 associate, map quality, 或 SQE

本文原发于应届生 BBS，发布时间：2011 年 2 月 23 日

地址：<http://bbs.yingjiesheng.com/thread-863242-1-1.html>

2 月 22 号，不是我的幸运日。我收到了 google 的 thank you letter。

等待一周等来这个结局，未来又开始变得虚无飘渺起来。

收到邮件后打了个电话过去，想争取，却改变不了事实，人事说欢迎以后再再来应聘，她邮件里也说了，保留简历一段时间，如有匹配再联系。

想想也是，人生波折不断，我怎么可能如此幸运，一毕业就乘上自己梦想的公司。

在给这次应聘划上句点的同时，也整理下自己面试经历，以惠及后人，也算给 google 做个宣传，在应届生网上可是很久不见 google 面试的更新，以至于大家都怀疑 google 是否还存在。

本人背景：女生，本科软件工程，硕士国外计算机工程，有软开工作经验 1 年及其他实习经验，内推。

应聘职位：map quality associate

面试：2 月 15 日 4 轮一对一面试，每轮半小时，每个面试官针对不同角度对你进行提问。

面试官都是 SQE 队伍的，也就是 search quality evaluator 团队。每个人风格也不一样，第一个很亲和；第二个一个问题快问下去，有点像压力测试，对方做销售市场的；第三个是工程师，不是技术 面试也就很轻松，对着简历问下去；第四个感觉是个头头，聊天随便问。

1) 自我介绍，职业定位，对 google 产品的了解，竞争对手的相应产品及其优缺点，平时上网浏览什么网页，情景模式（如果你是风投，投资哪家互联网公司；如果过去两年重来，会做什么；如果政府授权你改善交通，你有什么主意等等），你有什么问题。

2) 最体现你解决问题，分析问题能力的事件，针对这个事件使劲问。还问你如何做 presentation 的事宜。最后问你有什么问题。

3) 过简历，对 google map 的改进，和竞争对手的产品比较，然后对搜索了解吗。

4) 会编程吗，http 访问协议等等，因为工作还是需要编程写点代码的。遇到最大的挫折是什么。

问题都不难，只要自信，对 google 产品及行业有一定了解，就能表现得不错。相对其他公司某些为了刷人而刷人的笔试，面试等，这还是较简单的。也或许我没看出简单里的难点，所以才失败了。

邮件里写的是，面试对我印象不错，可是仔细看过简历，觉得没有特别匹配。确实，简历只是列举经历成就等，并没针对这个职位的需求做特别描述。

和内推的朋友说起，他说我表现得太想进 google 了，甚至为此做了个视频，一方面是好事，一方面却给人事扣分机会。人事也怕招错人，所以会比较谨慎。你这么喜欢 google 还告诉她手上还没有其他 offer，那不是绕一圈可能还会回去。她没必要现在就把你招进来。

不管怎么说，错过了这次机会，很遗憾。写此文悼念。

3.11 Google TAM 面试记

上周四 9 号面试

周三晚上去北京，结果航空管制晚点 2 小时，还以为棒子打仗了。。到同学家已经 2 点了。被出租车绕圈绕了 94 块。。

15 点开始面试

半小时笔试，主要网络方面，css，html 啥的，做的很差
接下来 2 小时 4 个面试官车轮战，每个半小时，有几个问题介绍下

- 1.请分析下年底股市走势（看我对经济感兴趣）
- 2.说得好，下面反着说一下（。。。）
- 3.你最成功的一件事
- 4.你择业的标准
- 5.急转弯，如何让 html 里一句话不被用户看到

还有很多问题忘记了，就是谈简历的，气氛还是很愉快的
bless。。

3.12 谷歌面试问题

据国外媒体报道，在当前经济形势不景气的情况下，谷歌招聘新员工是一件令人振奋的事，特别是对那些在当前金融风暴中渴望找到安全港的年轻经理们和软件开发商们来说是个好消息。不过，也不要高兴太早，谷歌在招聘新员工时，更加青睐名牌大学的学生，即便你是人到中年，招聘时谷歌也会看你的大学平均成绩。谷歌需要的是那些有志改变世界的具有远大抱负的人才。不仅如此，即便是你达到了它们所有的这些要求，你还必须通过谷歌的面试，而谷歌面试的问题真可谓是五花八门，以下是挑选的 15 个应聘不同职位所遇到的让你摸不着头脑的面试问题。

- 1、一辆校车可以装多少个高尔夫球？

应聘职位：产品经理。

- 2、如果让你清洗西雅图的所有窗户，你会要求多少报酬？

应聘职位：产品经理

3、某个国家的人们都仅想生养男孩……，所有家庭在生养男孩之前是不会停止生养的。如果生养的是女孩，他们将继续生养，只到生个男孩为止。如果生养的是男孩，他们将不再生养。那么，在这个国家男孩和女孩的比例是多少？

应聘职位：产品经理

- 4、全世界有多少位钢琴调音师？

应聘职位：产品经理。

- 5、为什么井盖是圆的？

应聘职位：软件工程师

- 6、如何为旧金山市制定人员疏散计划？

应聘职位：产品经理

- 7、每天时针和分针重合多少次？

应聘职位：产品经理

- 8、“死牛肉”的意思是什么？

应聘职位：软件工程师

- 9、把车停在旅馆，财物丢失，接着会发生什么？

应聘职位：软件工程师

10、你需要确认一下朋友鲍勃是否有你正确的电话号码，但不能直接问他。你须在一张卡片上写下这个问题，然后交给伊夫，由伊夫把卡片交给鲍勃，再转告你答案。除了在卡片上写下这个问题外，你还必须怎样写，才能确保鲍勃在给出答案的同时，不让伊夫知道你的电话号码？

应聘职位：软件工程师

11、你是一艘海盗船的船长，船员开始投票表决如何分配掠夺的赃物。如果支持你提出分配方法的海盗不到一半，你就必须被处死。你将提出怎样的分配方法，才能确保你在分到应得赃物的同时，不会被处死？

应聘职位：工程经理

12、有八个大小相同的球，其中 7 个重量相等，有一个稍微重一点。如何用天平仅称两次就能确定哪个球更重？

应聘职位：产品经理

13、有两个蛋，有一幢 100 层高的楼。蛋要么很坚固，从 100 层楼摔下都不会被摔破，要么很脆弱，从一层楼摔下，都可能被摔破。两个蛋外形相同。计算一下一只蛋最高可从这幢 100 层楼的哪一层楼摔下而不会被摔破。问题是你需要摔几次才能计算出来。在这个过程中，你可以摔破这两只蛋。

应聘职位：产品经理

14、如何用三句话向你 8 岁大的侄子解释什么是数据库？

应聘职位：产品经理

15、你被缩小到五分钱那么高，这是按比例缩小的。之后，你被放到一个空的透明搅拌器中。搅拌刀片将在 60 秒后转动。你将怎么办？

应聘职位：产品经理。

3.13 谷歌疯狂面试题

A friend of mine had an interview a couple weeks ago with Google Inc. He provided me a list of just some of the questions he was asked. I've added a few more from others I have talked to who had interviews with the internet giant, Google, as well. See if you can answer them. Many are open ended with several right answers, therefore I did not provide the answers.

1. How many golf balls can fit in a school bus?
2. You are shrunk to the height of a nickel and your mass is proportionally reduced so as to maintain your original density. You are then thrown into an empty glass blender. The blades will start moving in 60 seconds. What do you do?
3. How much should you charge to wash all the windows in Seattle?
4. How would you find out if a machine's stack grows up or down in memory?
5. Explain a database in three sentences to your eight-year-old nephew.
6. How many times a day does a clock's hands overlap?
7. You have to get from point A to point B. You don't know if you can get there. What would you do?
8. Imagine you have a closet full of shirts. It's very hard to find a shirt. So what can you do to organize your shirts for easy retrieval?
9. Every man in a village of 100 married couples has cheated on his wife. Every wife in the village instantly knows when a man other than her husband has cheated, but does not know when her own husband has. The village has a law that does not allow for adultery. Any wife who can prove that her husband is unfaithful must kill him that very day. The women of the village would never disobey this law. One day, the queen of the village visits and announces that at least one husband has been unfaithful. What happens?
10. In a country in which people only want boys, every family continues to have children until they have a boy. if they have a girl, they have another child. if they have a boy, they stop. what is the proportion of boys to girls in the country?
11. If the probability of observing a car in 30 minutes on a highway is 0.95, what is the probability of observing a car in 10 minutes (assuming constant default probability)?
12. If you look at a clock and the time is 3:15, what is the angle between the hour and the minute hands? (The answer to

this is not zero!)

13. Four people need to cross a rickety rope bridge to get back to their camp at night. Unfortunately, they only have one flashlight and it only has enough light left for seventeen minutes. The bridge is too dangerous to cross without a flashlight, and it's only strong enough to support two people at any given time. Each of the campers walks at a different speed. One can cross the bridge in 1 minute, another in 2 minutes, the third in 5 minutes, and the slow poke takes 10 minutes to cross. How do the campers make it across in 17 minutes?

14. You are at a party with a friend and 10 people are present including you and the friend. your friend makes you a wager that for every person you find that has the same birthday as you, you get \$1; for every person he finds that does not have the same birthday as you, he gets \$2. would you accept the wager?

15. How many piano tuners are there in the entire world?

16. You have eight balls all of the same size. 7 of them weigh the same, and one of them weighs slightly more. How can you find the ball that is heavier by using a balance and only two weighings?

17. You have five pirates, ranked from 5 to 1 in descending order. The top pirate has the right to propose how 100 gold coins should be divided among them. But the others get to vote on his plan, and if fewer than half agree with him, he gets killed. How should he allocate the gold in order to maximize his share but live to enjoy it? (Hint: One pirate ends up with 98 percent of the gold.)

Do you still think you have what it takes to work for Google?

3.14 Google 面经小全

周围同学的一些 Google 的电话面经:

1. 项目经历;
2. 你觉得哪个项目最富有挑战性? 你怎么解决那些问题的?
3. struct 和 class 的区别
4. linux 下的一些基本命令
5. 引用和指针的区别, 用法
6. 单向链表的删除操作, 已知 head,p(指向被删除元素), 要求复杂度为 $O(1)$ (题目似有误)
7. Google Desktop Search 的一些技术法? 后面的问题更有一些深度: 这个算法(他会给出一个正确的算法思路)有什么效率上的问题, 如何解决; 如果这个算法经常要被调用, 如何设法使效率提高?
9. 对现在的 Stack (栈) 数据结构进行改进, 加一个 min() 功能, 使之能在常数, 即 $O(1)$, 时间内给出栈中的最小值。可对 push() 和 pop() 函数进行修改, 但要求其时间复杂度都只能是 $O(1)$ 。
10. 用 C/C++ 编程如何确定所在的计算机上栈的增长方式 (是从高到低, 还是从低到高)
11. 你要如何实现类似 Google 的拼写检查 (即纠正用户输入关键字中的错误单词)?
12. 如果进入 Google, 让你自由地选择一个课题, 你会做什么方面的?

3.15 google 面试试题详解

Google 考题分为几大类: 日常知识型、思考型。还有一些我们甚至不知道用意是什么, 凭什么拿来做面试题... 也许在我们一起研究的同时, 可以得出一些结论, 如果你知道某个题目的答案或者有自己的看法、见解直接在下面留言, 我将总结到文章中去:

一辆学校班车里面能装多少个高尔夫球?

答: 应该也是用常理推断过程

你被缩小到只有硬币厚度那么点高（不是压扁，是按比例缩小），然后被扔到一个空的玻璃搅拌器中，搅拌刀片一分钟后就开始转动。你怎么办？

答：搅拌器应该是有空隙的，所以躲到边上应该不会被打到。但是玻璃搅拌器四周可能无法抓住附着，所以旋转带来的风可能把你吹起来。所以尽量走到搅拌器转轴中间，试图爬上去或者抓住。

要是让你清洗整个西雅图的所有窗子，你会收取多少费用？

答：类似调音师的推理过程

怎么才能识别出电脑的内存堆栈是向上溢出还是向下溢出？

答：只能向上溢出

你要向你 8 岁的侄子解释什么是数据库，请用三句话完成。

答 1：数据库就如存钱罐…

答 2：就是你的书包，里面有你喜欢：圣斗士金卡，小玩具；也有你不喜欢的：考卷啊，要家长签名的东西啊。。。反正里面各种各样的东西都有，但绝大多数可能都不是你放进去的，但你却要注意收拾。

时钟的指针一天内会重合几次？

答：如果是没有秒针且分针不是按 1 分钟递进的那种钟表，那么可以重合多次（22 次吧），如果是按分钟递进的或者有秒针的，那就重合两次。另外，还要考虑齿轮的齿距和制表匠的水平。因此从微观上讲，那两根或三根针的很难重合。。。。。。

你需要从 A 地去 B 地，但你不知道能不能到，这时该怎么办？

答：以目前科学水平，只要 A 地 B 地都叫得出名字并且都在地球表面的陆地上，都可以到。

好比你有一个衣橱，里面塞满了各种衬衫，你会怎么整理这些衬衫，好让你以后找衬衫的时候容易些？

答 1：优先颜色，其次款式，再次新旧程度

答 2：按季节、场合、性别分

有个小镇有 100 对夫妇，每个丈夫都在欺骗他的妻子。妻子们都无法识破自己丈夫的谎言，但是她们却能知道其他任何一个男人是否在撒谎。镇上的法律规定不准通奸，妻子一旦证明丈夫不忠就应该立刻杀死他，镇上所有妇女都必须严格遵守这项法律。有一天，镇上的女王宣布，至少有一个丈夫是不忠的。这是怎么发生的呢？

答 1：全部男人都被杀死

答 2：国王被杀死了（可能女王也被杀死，这样才能确保秘密不会泄露）

在一个重男轻女的国家里，每个家庭都想生男孩，如果他们生的孩子是女孩，就再生一个，直到生下的是男孩为止。这样的国家，男女比例会是多少？

答：1:1/50%

如果在高速公路上 30 分钟内到一辆车开过的几率是 0.95，那么在 10 分钟内看到一辆车开过的几率是多少（假设为常概率条件下）

答 1： $1-(1-x)(1-x)(1-x)=0.95$ ，解出 x 就可以了，嘿嘿

答 2：0.95

答 3： $12 \text{ 度} * 0.25 = 3 \text{ 度}$

如果你看到钟的时间是 3:15，那一刻时针和分针的夹角是多少？（肯定不是 0 度！）

答：7.5

4 个人晚上要穿过一座索桥回到他们的营地。可惜他们手上只有一支只能再坚持 17 分钟的手电筒。通过索桥必须要拿着手电，而且索桥每次只能撑得起两个人的份量。这四个人过索桥的速度都不一样，第一个走过索桥需要 1 分钟，第二个 2 分钟，第三个 5 分钟，最慢的那个要 10 分钟。他们怎样才能在 17 分钟内全部走过索桥？

答 1：1+2 先过，1（或 2）返回，5+10 过，2（或 1）返回，1+2 过

答 2：最慢的 10 分钟在桥头打手电筒，1 分钟和 2 分钟先过，在 1 分钟过完时，5 分钟立刻上桥。在 2 分钟过完时，10 分钟拿着手电筒上桥，总共只花了 12 分钟就能全部过去

答 3：

先 1 分钟和 2 分钟的去，2 分钟呆在那边，1 分钟的回来，用了 $2+1=3$ 分钟了；

5 分钟和 10 分钟一起过去，2 分钟的回来，用来 $3+10+2=15$ 分钟了；

1 和 2 分钟最后一起过去，用了 $15+2=17$ 分钟了。

你和朋友参加聚会，包括你们两人在内一共有 10 个人在场。你朋友想跟你打赌，说这里每有一个人生日和你相同，你就给他 1 元，每有一个人生日和你不同，他给你 2 元。你会接受么？

答 1：这个题目好像有陷阱，首先自己肯定和自己生日相同，所以开始你就要给对方 1 元。然后剩下 9 个人里面，你需要有 4 个人和你生日不同，你才能赚回来。而 9 个人里面同时有 5 个人生日和你相同的概率我觉得是比较小了，所以换做我，我会接受的！

答 2：不接受

全世界有多少个钢琴调音师？

答 1：2 个，一个男的一个女的

答 2：对客户来讲就一个，因为所作的工作一样，所以统统可以外包掉

你有 8 个一样大小的球，其中 7 个的重量是一样的，另一个比较重。怎样能够用天平仅称两次将那个重一些的球找出来。

答 1：先取 6 个，天平上一边 3 个，同重则称剩余 2 个即可；不同重，则取重的 3 个中的 2 个来称。

答 2：将 8 个球按个数 2, 3, 3 任意分为三组：A、B、C。

将 B、C 两组分别置于天平两端，若两端持平，即质量相等，则只需将 A

组的两个球分别置于天平两端，向下倾斜的一端所盛的球即是比较重的；若两端倾斜，则将向下倾斜的一端所盛的 3 个球取出，再从这 3 个球中任意取出两个球分别置于天平两端。如果两端持平，那么未被抽取的那个球就比较重的；如果两端倾斜，那么向下倾斜的一端所盛的球即是比较重的；

答 3：3-3-2 分称

有 5 个海盗，按照等级从 5 到 1 排列。最大的海盗有权提议他们如何分享 100 枚金币。但其他人要对此表决，如果多数反对，那他就会被杀死。他应该提出怎样的方案，既让自己拿到尽可能多的金币又不会被杀死？（提示：有一个海盗能拿到 98% 的金币）

答 1：98, 0, 1, 0, 1

答 2：如果是我。。。我会提出让等级比我低的人继续按这个方法协商如何分，这样可以陷入逻辑悖论。只要完全按这个规则，那我就死不掉。。。。。

3.16 Google 员工出书大谈公司面试心得

Google 台湾一名员工日前出书，就自己的成长历程进行了回顾。其中谈到面试进 Google 的部分心得，总结为“13 绝招”。

这名员工 shi Google 台港业务总经理张成秀，她认为，尽管 Google 一直号称只要最好的人，并且进入公司前常常是多达 10 多道面试程序。她本人就是在 13 次面试后方才进入。但其中也会有很多技巧可以帮助通过面试。

这些技巧分别是：

一、前几轮面试争取改为视频

Google 面试前几关通常采用电话，这对很多英语非母语的人来说比较困难。很多优秀的人才在这一环节就惨遭淘汰。因此，不如询问 Google 改用视频方式，这样辅助动作，一些意思可以得到更有效表达。

二、化被动为主动

把自己对于应聘职位的想法写成企划案事先送给对方，提前掌握面试主导权。

三、事先了解面试官的情况

提前熟悉了解面试官的情况，显然是拉近距离的不错法子。

四、注重第一印象

要在众人中脱颖而出，第一印象扮演关键角色，比如加上自己写的一些好文章，能让老板对自己多些了解和

加深印象。

五、为他人设想

面谈前夕，好不容易才把所有的数据整理好，虽然也把电子文件寄过去了，但如果能把这些资料打印好送过去是不是更好？作者就是这么做的。不过，人人都送一份，那 Google 的老板就真的有的瞧了。

六、提前做好功课

提前做好功课，包括了解自己的职位，了解 Google 公司及它的产品总是不会错的。

七、个别案例分析

网上找数据，然后得出分析并不难。不过，针对个案的分析才是考验功夫的地方。

八、善用辅助工具

面谈 Google 会碰到很多棘手的问题，这时拿出纸笔或在黑板上画出来会更好些。

九、先讲结论或解决方案

面试时，记得先给出总结，再去分别讲述每一个主题。记住，每段的第一句就是重点。这个中文的表述习惯很不相同。

十、注意细节

你不会到 Google 面试却连 Gmail 都不知道吧？用他的产品或服务显然就是一些小细节。

十一、放张自己的个性图片

个人简历上来张个性照片，不过，强调个人特征或特色的插图显然更会让人记忆深刻。

十二、表达感谢

每关面谈完后，发封电子邮件给主考官表示感谢。有时没要到对方的联络方式，就请人事部门转寄。这是一种礼貌，更是一种纪律的表现。

十三、提前熟悉面试场地和环境

有时 Google 面试会约在公司外面，提前过去看看准没错。要是在公司的话，那就要提前找好行车路径了。

第四章 Google 求职综合经验

4.1 我的 Google 求职随感

我于两个月前开始申请 Google 的 Software Engineer，几天前拿到 Offer。由于签了保密协议，只好写写随感了。见谅。

9 月底在版上看到招聘消息，赶紧按照 Google 主页上提供的招聘信息用 Email 投递了简历。

两周过后收到 HR 的信，问我愿意什么时间面试。后面的几次面试时间也都这样商量决定，挺人性化的。

10 月中得到了第一次电话面试。面试官很和蔼，中途还问我要不要休息一下。我在讲述自己的想法时面试官不断地问我一些细节的问题。当时比较傻，忙着应付答案，挂了电话才发现这些问题其实是在帮助我找到更好的解决方案，提示一些新的思路。面试下来我还有点小小的领悟：数据结构和算法的知识并不只是拿来套用的，更高的要求是在面对新的问题时根据这些（数学）思想设计自己的算法。

1 周后得到现场面试的机会。面试双方穿着都很随意，而且有趣的是整个上午我始终坐在一个地方，而不同面试官依次进出房间对我进行面试。面试的气氛比较轻松，基本上处于交谈和讨论的过程中，遇到感兴趣的话题我甚至会反问面试官一些问题。技术面试时要进入状态，不妨假想自己坐在这里是在帮助面试官解决工作中的问题，而不是来回答问题，这样整个过程自己会主动一些。

面试使我对 Google 有了更直接的认识。面试我的几位工程师给我留下了很深的印象，而且整个过程中不经意反映出的公司内部的气氛也让我感到十分舒适。于是自己暗下定决心把 Google 放在求职意向的第一位。

又过了两周多，我收到来自另一个 HR 的信，信中说余下的流程由她协助我完成。收到这封信后，我才开始

意识到其中蕴含的可能性，前段时间稳定的心态顿时被打乱了。由于时差的关系，最新的消息通常是在深夜发过来的，所以那段时间睡觉也不安心。

紧接着第三次面试一反前面几位面试官的和蔼，每个问题都咄咄逼人。我发现最难对付的面试是整个过程不给你任何反馈，不管回答得好还是不好对方都没有任何情绪上的反应，而是紧接着问下一个问题，给人一种在向错误的方向上越走越远的感受。最后挂电话前我紧张得连谢谢、再见都忘了说了。

在接下来的半个多月的等待中，上面提到的那位 HR 一直和我保持着联系，及时告诉我最新的进展情况，给予我很大的支持。这段时间过得糊里糊涂，倒是自己的英文书信能力有了不小的提高。直到一天清早，我正在网上填另一个公司的申请表时，突然接到 HR 的电话：“Hi, we have some good news for you...”

另外我想谈谈自己在求职过程中的一些想法。相比很多人而言我的求职经历可以说是少得可怜，权当班门弄斧了：

(1) 自我定位。由于自己喜欢亲手做具体的东西，不喜欢搞纯研究，不喜欢管人或纯粹与人打交道，又没有别的特长，所以从一开始就有明确的职位定位——做一名程序员。求职路漫漫，最好不要在这段时间放弃自己在专业技能上的成长。

(2) 工作地点。由于我家在深圳，刚开始时一心想回去，所以一遇到非深圳的公司就犹豫再三。幸好及时转变了想法，把地点延伸到其它几个城市，才得以打开了局面。

(3) 认真应聘。一个公司的应聘过程并不是从面试开始，而是从你填申请表的第一个字就已经开始了。所以我觉得在精力有限的情况下，应当坚持应聘的质量而不是数量。海选的随机因素太多，如果可能应该尽早动用各种途径来争取面试机会，就算没有也应保持关注、主动和 HR 或相关部门联系。

最后，祝愿还没拿到 Offer 的同学早日功德圆满。

4.2 我的找工作之路-google 篇

Google 可以说是我最喜欢和最想进去的一间公司，也是我第一间面的公司，所以决定详细写下。大三下的时候跑去投了 google 的实习，当时找了里面的师兄帮忙推荐，于是直接免掉了笔试，不过出于对 google 的无比仰慕，还是去听了宣讲，也跑去做了笔试，感觉 google 的笔试重视的是 CS 所学的基础知识，特别是选择题，操作系统，编译原理，编程等都考察到了，而且在网上搜笔经的时候有看到说其他人说起过如果笔试时候的选择题没达到要求的题目数就会被直接 cut 掉，不看后面的主观题，虽然无从验证消息的真实性，不过大家做笔试的时候还是要多加小心。然后是笔试的主观题一般似乎都是考的算法，对于平时没怎么写过算法题目的同学会比较难，不过我认识的一起去笔的 ACM 的大牛们都觉得很简单，所以建议以后想去笔 g 家的师弟师妹们还是要好好准备下算法这 part，g 还是蛮重视这方面的。

然后再然后就是面试，被叫了去东方宾馆面试，也是偶的处女面，当时心里超级紧张，签了保密协议就在那里等。

第一面面我的是 stephenGe，问了我一个很经典的算法题目的变种，属于该经典问题的一个特殊情况，不过由于签了保密协议就不在这里说了，如果有看过《具体数学》这本书的话应该有看过类似的变种，是有 $O(1)$ 的算法。可惜我当时真的是太紧张了，虽然看过类似的题目也想往那个方向想，可惜还是没能想出来，于是只能从头开始推，最后勉强在错了几次和面试官的提醒下，推出了个 $O(n)$ 的，然后就结束了。其实我想应该会面两道的，不过可能因为我的时间已经用完了，所以面试官没有再问。

然后当天回去后以为要当炮灰了，结果过了几天好很惊异的发现自己收到了二面通知，当时真的是很开心，估计是因为我的答案虽然没有最优，但是 $O(n)$ 对那个经典问题的一般情况可以算是最优的，所以被放过了吧。然后二面面我的是 wang chaogg，很 nice 的一个人，问的问题也不难，方法一下子就想到了，不过我还是经验不足，写代码的时候写出了很多小错误，影响了这面的成绩。

这里提醒一下大家，要面 google，微软这类可能会要求你当场写出可运行代码的公司的时候，最好在去面试

之前好好练习一下，先自己习惯一下在纸上 coding 的感觉，我就是训练得太少，结果当时出了很多低级错误，例如把 $a \leq 1$ 写成了 $a < 1$ 之类，搞得每次改完一编 wang chao 都会问我，"你要不要再检查一下"，然后还每每能找出我的错误，弄得我当时完全郁闷了...还有就是 google 对细节的要求是很高的，记得当时我代码当中写了一句 $\text{int } n = \lg(s) / \lg(2)$ ；然后面试官就当场就把我的代码敲了进去测试，把结果给我看，原来是想告诉我这句可能会有精度问题，话说我虽然平时也知道会有这种问题存在，可是没想到会要求到这个程度，还好我接着说那就加个 $1e-7$ 之类的就好了，似乎面试官觉得这个答案还可以，没继续追问...然后再然后就拿了 google 的实习 offer，去了北京。这里再说一下实习里面学到的东西，好的 coding style，这是我之前比较欠缺的一方面，也是我本科很少训练的一方面，之前都是觉得写个程序能把结果弄出来就是了，其实一个好的 coding style 会使你的代码易读性大幅度提高。

然后是文档和注释的规范化，在那里第一次体验了写文档和写注释比写代码花费更多时间的情况，无数次被老板让我改注释，话说我之前写程序都是很少写文档的...同时也很佩服 engineer 们对代码的严格要求，严格到了就算注释里面少打了一个空格都会很细致的给我指出，有时也许会想，也许就是这些对无数性能和细节上的精益求精，才造就了 google 今天的成功吧~ 写出这些是希望师弟师妹们吸取我的教训，在 coding 的时候也注意下这些方面~

再后来就是 intern 的 conversion，也就是申请 full time 再加的两轮面试，不过其中一面面得不好，于是 fail 掉了。最后再介绍下我了解到的 google 的面试制度，一般申请 full time 最少要面 4 面，由 4 个不同的 engineer 来面，面试的内容和范围不定，不过一般都会根据你简历上写的来问，所以如果自己没做过的项目，不擅长的方面就不要往上写了...如果你是本科生没做过什么项目，又除了学校教的一些基础知识外没学过多少课外的东西，那么一般一开始都会被问到算法，我自己碰到的难度感觉上大概是 poj 上那些 3, 4 百人过的难度，不过在水木看到有人说碰到的难度和 top coder srm div1 的 250 分题目差不多，这个难度对那些平时搞算法的大牛估计就是秒杀的难度了，不过如果是平时比较少玩算法的，可以针对上面的难度准备下~

附录：更多求职精华资料推荐

强烈推荐： [应届生求职全程指南（第十二版，2018 校园招聘冲刺）](#)

下载地址： <http://download.yingjiesheng.com>

该电子书特色： 内容涵盖了包括职业规划、简历制作、笔试面试、企业招聘日程、招聘陷阱、签约违约、户口问题、公务员以及创业等求职过程中的每一个环节，同时包含了各类职业介绍、行业及企业介绍、求职准备及技巧、网申及 Open Question、简历中英文模板及实例点评、面试各类型全面介绍、户口档案及报到证等内容，2018 届同学求职推荐必读。

应届生求职网 YingJieSheng.COM，中国领先的大学生求职网站

<http://www.yingjiesheng.com>

应届生求职网 APP 下载，扫扫看，随时随地找工作

<http://vip.yingjiesheng.com/app/index.html>

