
Facebook面试及准备指南



本资料由九章算法独家倾情整理

获取更多干货，欢迎关注**九章算法**微信公众号



九章算法，专业的 IT 求职面试培训，团队成员均为硅谷和国内顶尖IT 企业工程师。

目前开设课程有：[《九章算法班》](#) [《系统架构设计》](#) [《九章算法基础班》](#) [《Java+Python》](#) [《面试软技能指导》](#) [《九章算法高频题冲刺班》](#) 等。

大厂真题任意刷，欢迎关注 **LintCode** 微信公众号



全球名企面试真题汇总，剖析当前科技行业热点动向，算法与人工智能专业训练，LintCode 助你斩获理想 Offer！

Facebook (Meta) 编码面试真的很有挑战性。这些问题很困难特定于 Facebook，并且涵盖了广泛的主题。

好消息是，正确的准备可以产生很大的不同。我们分析了 Facebook 候选人报告的 170 多个软件工程师面试问题，以确定最常见的问题类型。下面，我们提供了一个精选的真实示例问题列表，包括免费解决方案。

此外，您还可以找到最佳资源的准备技巧和链接，以便您可更有策略地进行准备，并最大限度地提高在 Facebook 担任软件工程师工作的机会。

以下是我们将涵盖的内容的概述：

流程和时间表

示例问题

- 编码面试
- 系统设计面试
- 行为面试

准备技巧

请注意，如果您申请的是工程经理职位，[本指南](#)将更适合您。我们还针对[数据工程师](#)、[前端工程师](#)和[生产工程师](#)的角色提供了[具体指南](#)。

一、面试流程及时间安排

1.1 期待什么样的面试

Facebook 软件工程的面试流程和时间表是什么？平均需要四到八周，并遵循以下步骤：

1. 简历、求职信和推荐信
2. 招聘人员电话屏幕：一次面试
3. 技术电话筛选：一到两次面试
4. 现场：四到六次面试

让我们在下面更详细地了解这些步骤中的每一个：

1.1.1 简历、求职信和推荐信

第一步是首先接受 Facebook 采访。在本指南中，我们主要关注采访，因此我们将保持这部分的简短。您可以直接向 Facebook 申请，或者招聘人员可以通过 LinkedIn（或类似网站）与您联系。在任何一种情况下，拥有一份适合前端职

位的高质量（和最新的）简历都是有帮助的，更具体地说，它是为 Facebook 量身定制的。

如果您确实与 Facebook 中的某个人有联系，将员工推荐给内部招聘团队也可能会有所帮助，但如果您无法做到这一点请不要担心。

1.1.2 招聘人员手机屏幕

在大多数情况下，您将通过与人力资源招聘人员通电话来开始 Facebook 的面试过程。他们希望确认您是否有机会获得这份工作，因此请准备好解释您的背景以及您为什么适合 Facebook。您应该期待典型的行为和简历问题，例如“介绍一下你自己”、“为什么选择 Facebook？”或“介绍一下您作为开发人员的日常工作”。

如果您通过了第一个 HR 筛选，招聘人员将帮助您与 Facebook 工程师安排您的第一个技术筛选。Facebook 的一大优点是他们的招聘过程非常透明。因此，您的人力资源联系人将引导您完成招聘过程中的剩余步骤，并将与您分享一个有用的电子邮件列表资源，您可以使用这些资源进行准备。

1.1.3 技术电话屏幕

您将经历一两个最初的 45 分钟技术屏幕，这些屏幕通常通过电话进行。你的面试官会从一些行为问题开始，但大部分时间会花在典型的编码问题上（例如数据结构、算法等）

对于这些电话面试（例如 CoderPad），您将使用一个没有语法高亮或自动完成功能的简单在线代码编辑器，并且最好事先习惯使用其中一个。

最后，如果你住在 Facebook 办公室附近，并且真的有加入公司的动力，你可以要求到现场进行初步筛选。您的招聘人员通常会很乐意为您安排此活动，因为这对您和您的面试官来说都是更好的体验。

1.1.4 现场访谈

现场面试才是真正的考验。你通常会在 Facebook 办公室度过一整天，总共进行四到六次采访。每次采访将持续约 45 分钟 涵盖以下主题之一：

1. **编码面试**，您将在其中解决一般的编码问题，这些问题通常比最初的技术屏幕更难。
2. **系统设计面试**，根据你的背景，你会被问到一个系统或产品设计问题

-
3. **行为面试**，您可以在其中期待有关您的背景、成就和在 Facebook 工作的动机的问题。

你通常会得到三场编码面试、一场设计面试和一场行为面试。但正如我们将在下一节中讨论的那样，确切的细分将根据您申请的角色、团队和级别而有所不同。

除了这些采访之外，您还可以在现场与一位工程师同事共进午餐。午餐面试的目的是让您有时间询问在 Facebook 工作的感受。在此期间，公司不会对您进行评估，但我们建议您表现得像他们一样。

1.2 级别和角色的区别

我们上面描述的过程会根据你面试的级别和角色而略有不同。以下是您应该预料到的一些最常见的差异。

1.2.1 初级与高级软件工程师

对初级工程师和高级工程师的期望是不同的。

如果您是入门级/初级工程师，您将有望在编码面试中表现出色。但是你通常只会进行一次系统设计面试，其绩效标准会低于更高级的候选人，如果它与你的角色无关，你可能不会接受任何设计面试。

如果你是一名高级工程师，那么你们都需要在编码和系统设计面试中表现出色。在某些情况下，您还可以进行两次设计面试（例如，您正在申请基础架构团队）。

1.2.2 个人贡献者与管理者

Facebook 有两条职业道路。您可以成长为**经理**，最终领导工程师团队（管理轨道）。或者，您可以在技术上保持亲力亲为，并随着您变得更资深（个人贡献者轨道）而专业化。

如果您以个人贡献者的身份进行面试，那么您应该主要期待技术面试，并且通常只会进行一次行为面试。

如果你以经理、主管或以上的身份面试，那么你应该期待至少两次行为面试。准备好回答有关您如何发展人员、与跨职能团队合作、执行项目、发展组织等方面的问题。

1.3 幕后发生的事情

您的招聘人员正在领导这一过程，并将您从一个阶段带到下一个阶段。以下是上述每个阶段的幕后情况：

- **在技术电话屏幕之后**，与您交谈过的面试官有 24 小时的时间向内部系统提交他们的评分和备注。然后，您的

招聘人员会查看反馈，并根据您的表现决定是否将您转到现场面试。

- **在 Onsite 之后**，四到六位面试官会提出是否聘用你的建议，招聘人员会整理你的“资料包”（面试反馈、简历、推荐信等）。如果他们认为你能得到这份工作，他们会展示你的下次候选人审查会议上的情况。
- **候选人审查会议**用于评估所有最近完成面试循环并接近获得报价的候选人。将对您的数据包进行分析，并讨论可能存在的问题。你的面试官被邀请参加你的候选人审查会议，但通常只有在你收到的成绩有很大分歧的情况下才会参加（例如，2 次没有录用，2 次录用）。如果在讨论之后，团队仍然无法就你是否应该得到 offer 达成一致，你可能会被要求进行后续面试以解决争论。在候选人审查会议结束时，将提出聘用/不聘用建议供招聘委员会考虑。
- **招聘委员会**包括来自 Facebook 的高级领导。这一步通常是一种形式，委员会遵循候选人审查会议的建议。主要重点是微调确切的水平，因此将为您提供补偿。

同样重要的是要注意，招聘经理和推荐你的人对整个过程的几乎没有影响。他们可以帮助您在开始时获得面试，但仅此而已。

2. 示例问题

在 IGotAnOffer, 我们相信数据驱动的面试准备, 并使用 [Glassdoor](#) 数据来确定 Facebook 最常问的问题类型。

对于编程面试, 我们将您将被问到的问题细分为子类别 (例如数组/字符串、图形/树等), 以便您可以在准备中优先考虑最常见的问题。对于系统设计和行为问题, 我们列出了 [Glassdoor](#) 上经常报告的 10 个示例问题。

2.1 编码面试

www.jiuzhang.com

2

Facebook coding interview questions

#1 Arrays / Strings

E.g. Product of array except self

#2 Graphs / Trees

E.g. Range sum of a binary search tree

#3 Dynamic Programming

E.g. Continuous subarray sum

#4 Search / Sort

E.g. Find the minimum number of meeting rooms required

#5 Linked lists

E.g. Copy list with random pointer

#6 Stacks / Queues

E.g. Implement Queue using Stacks

Source: Glassdoor.com data for Google PM interview questions. Analysis by IGotAnOffer.com

Facebook 软件工程师解决了公司在代码方面面临的一些最困难的问题。因此，他们必须具备强大的解决问题的能力。面试的一部分，您希望展示您以结构化的方式思考并编写 准确、无错误且快速的代码。

以下是 Facebook 编码面试中最常见的问题类型及其频率。请注意，下面的列表不包括我们在下一节中介绍的系统/产品设计和行为问题。

-
1. [数组/字符串](#) (38% 的问题, 最常见)
 2. [图表/ 树](#) (29%)
 3. [动态规划](#) (18%)
 4. [搜索/排序](#) (9%)
 5. [链表](#) (4%)
 6. [堆栈/队列](#) (2%, 最不频繁)

我们还在下面列出了 Facebook 用于这些不同类型问题的常见示例。为了使这些问题更容易学习, 我们修改了措辞以匹配 Leetcode 或其他资源上最接近的问题, 并且我们已经链接到一个免费的解决方案。

最后, 我们建议阅读我们的指南, [了解如何回答编码面试问题](#) 以了解更多关于解决这些问题的分步方法。

Facebook 提出的示例编码问题

1. [数组/字符串](#) (38% 的问题, 最常见)

- “给定一个 $n > 1$ 的整数数组, 返回 $nums$ 一个数组, 使得它等于除的所有元素的乘积。” ([解决方案](#))
`output[i] = nums[i]`

-
- “给定一个非空字符串 s ，你最多可以删除一个字符。判断你是否可以把它变成回文。”（[解决方案](#)）
 - “实现 next permutation，它将数字重新排列成字典上的下一个更大的数字排列。”（[解决方案](#)）
 - “给定一个字符串 S 和一个字符串 T ，找到 S 中的最小窗口，它将包含 T 中的所有字符，复杂度为 $O(n)$ 。”（[解决方案](#)）
 - “给定一个字符串数组 $strs$ ，将字谜组合在一起。”（[解决方案](#)）
 - “给定一个字符串 s 只包含字符 '(', ')', '{, }', '[' 和 ']'，确定输入字符串是否有效”（[解决方案](#)）
 - “给定一个由 n 个整数组成的数组，是否有元素 a, b, c 使得 $a + b + c = 0$ ？在数组中找到所有唯一的三元组，其总和为零。”（[解决方案](#)）

2. 图表/树 (29%)

- “给定 $root$ 二叉搜索树的节点，返回所有节点的值之和，其值介于 L 和 R （含）之间。”（[解决方案](#)）

-
- “给定一棵二叉树，将其转换为循环双向链表（就地）。”（[解决方案](#)）
 - “在二叉搜索树（BST）上实现迭代器。您的迭代器将使用 BST 的根节点进行初始化。”（[解决方案](#)）
 - “给定一棵二叉树，你需要计算树的直径长度。”（[解决方案](#)）
 - “序列化和反序列化二叉树”（[解决方案](#)）
 - “给定一棵二叉树，求最大路径和。”（[解决方案](#)）
 - “给定一个外星语言的排序字典（单词数组），找出该语中字符的顺序。”（[解决方案](#)）
 - “检查给定图是否为二分图”（[解决方案](#)）

3. 动态规划 (18%)

-
- “给定一个非负数列表和一个目标整数 k ，编写一个函数来检查该数组是否有一个大小至少为2 且总和为 k 的倍数的连续子数组，即总和为 $n*k$ 其中 n 也是一个整数。”（[解决方案](#)）
 - “假设您有一个数组，其中第 i 个元素是给定股票在第 i 天的价格。如果您最多只能完成一次交易（即，买入并卖出一股股票），请设计一个找到最大利润的算法。”（[解决方案](#)）
 -
 - “给定一个输入字符串（ s ）和一个模式（ p ），实现支持 '.' 和 '*' 的正则表达式匹配。”（[解决方案](#)）
 - “给定一个非负整数列表 a_1, a_2, \dots, a_n 和一个目标 S 。现在您有 2 个符号 $+$ 和 $-$ 。对于每个整数，您应该从 $+$ 和 $-$ 中选择一个作为其新符号。找出有多少种分配符号的方法以使整数之和等于目标 S 。”（[解决方案](#)）

4. 搜索/排序 (9%)

-
- “我们有一个points平面上的列表。找到K离原点最近的点(0,0)。” ([解决方案](#))
 - “给定两个数组，编写一个函数来计算它们的交集” ([解决方案](#))
 - “给定一组由开始和结束时间组成的会议时间间隔[[s1,e1],[s2,e2],...] 找到所需的最小会议室数量。” ([解决方案](#))

5. 链表 (4%)

- “给出了一个链表，使得每个节点都包含一个额外的随机指针，该指针可以指向列表中的任何节点或为空。返回列表的[深层副本](#)。” ([解决方案](#))
- “给定一个单链表 $L: L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$ ，将其重新排序为： $L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$ ” ([解决方案](#))

6. 堆栈/队列 (2%)

- “使用堆栈实现队列的以下操作。” 注意：在以下链接中查看更多详细信息。 ([解决方案](#))

2.2 系统设计面试

Facebook、Instagram 和 Whatsapp 的月活跃用户均超过 10 亿。因此，Facebook 工程师需要能够设计具有高度可扩展性的系统，而设计面试由一到两次 45 分钟的对话组成，是大多数（但不是全部）软件工程师候选人面试过程的重要组成部分。

虽然您会发现大多数指南和资源通常只涉及“系统设计”面试，但 Facebook 会做出区分，并且通常会为您提供系统设计面试和产品设计面试之间的选择。

有什么不同？系统设计面试要求您处理非常大规模的分布式系统。考虑可用性、可扩展性、分区容错性、可靠性、存储、数据库、核心系统等。同时，产品设计面试可能更适合习惯于面向用户的产品候选人。想想 API、数据建模、客户端和服务端如何交互、用户如何与之交互等等。

无论你是面对系统还是产品设计面试，问题都会非常开放，感觉更像是一场讨论。因此，您将希望表明您可以同时具有创造力和结构。

如前所述，如果您是初级开发人员，您的期望将低于中级或高级开发人员。此外，对于某些角色（例如基础设施、安全等），您可能会接受多次设计面试。

以下是 Glassdoor 上 Facebook 采访报告中提出的最常见的系统和产品设计问题。有关更多信息，我们建议阅读本 [指南](#) 或[我们自己的指南](#)，了解如何回答系统设计问题，并在本[课程](#)中练习系统设计问题。

Facebook 提出的 10 大系统和产品设计问题

- 你会如何设计 Instagram / Instagram Stories?
- 你会如何设计 Facebook?
- 你会如何设计 Facebook Messenger?
- 您将如何设计 Facebook 对帖子评论的实时更新?
- 您将如何设计在线协作编辑器（例如 Google Docs）?
- 你会如何设计一个打字头功能（例如谷歌搜索自动完成）?
- 你会如何设计 Twitter 的热门话题?
- 您将如何设计分布式僵尸网络?

- 您将如何设计一个每小时可以处理数百万笔卡交易的系统？
- 您将如何从头开始为 Facebook 的企业网络设计安全性（安全团队采访）？

2.3 行为面试

Facebook 工程师通常与其他工程师、产品经理、设计师、数据分析师等在跨职能团队中工作。他们需要能够清晰地沟通、高效地与他人合作、建立信任和关系等。

因此，面试的行为部分（Facebook 称其为“了解你的面试”）是你真正想表明你擅长与他人合作的地方。正如我们之前提到的，如果您申请的是经理或以上职位，那么您将进行多次行为面试，并涵盖诸如领导团队、培养人才、管理日常事务等主题。

根据 Glassdoor 的数据，下面我们列出了 Facebook 倾向于提出的十个常见行为问题。有关更多信息，请查看我们关于如何回答行为面试问题和“为什么选择 Facebook？”的文章。题。

Facebook 提出的十大行为问题

- 说说你自己

-
- 为什么是脸书？
 - 告诉我最近/最喜欢的项目以及您遇到的一些困难
 - 告诉我你职业生涯中最大的成就
 - 告诉我你与一位同事一起工作时遇到的困难
 - 告诉我一次你必须解决团队中的冲突
 - 告诉我有一次你得到了建设性的反馈
 - 告诉我你不得不站出来为他人负责的时候
 - 告诉我你曾经在一个软件项目上苦苦挣扎的经历
 - 告诉我你最糟糕的老板以及他们为什么不好

3. 如何准备

既然您知道会出现什么问题，那么让我们专注于如何准备。

Facebook 的绩效标准很高，这已不是什么秘密。有些人甚至[辞掉工作](#)全职准备面试。

这显然是极端的，不是我们推荐的做法，但它显示了一些候选人准备投入多少努力。我们在下面列出了我们建议采取的四个步骤，以尽可能有效地进行准备。

3.1 了解 Facebook 的文化

大多数候选人都没有做到这一点。但在投入数十小时准备 Facebook 面试之前，你应该花一些时间来确保它确实是适合你的公司。

Facebook 享有盛誉，因此很容易完全忽略这一步。但根据我们的经验，声望本身不会让你每天都开心。这是工作的类型以及与您一起工作的人的意愿。

如果您认识在 Facebook 工作或曾经在那里工作的工程师，最好与他们交谈以了解文化是什么样的。此外，我们建议阅读 Facebook 的 [5 大核心价值观和黑客文化](#)。

3.2 自己练习

如上所述，您将在 Facebook 进行三种类型的面试：编码、系统设计和行为。准备的第一步应该是复习这些不同类型的 问题并练习自己回答。作为起点，请务必阅读 [Facebook 的软件工程师现场面试指南](#)。它充满了很好的建议，并为三种 面试类型中的每一种都提供了一些有用资源的链接。

对于**编码面试**，我们建议使用我们的[编码面试准备](#)文章作为您的一站式商店，以指导您完成准备过程。

对于**系统和产品设计面试**，我们建议学习我们的[系统设计面试准备指南](#) 并学习[如何回答系统设计面试问题](#)。这些指南涵

盖了回答系统设计问题（与产品设计同样相关）的分步方法， 并提供了几个带有解决方案的示例问题。

对于**行为访谈**，我们建议学习[我们的分步方法](#)来回答此类问题。此外，您还需要写下我们在上一节中列出的常见行为问题的答案。

最后，练习编码、系统设计和行为问题的一个好方法是大声采访自己。

这可能听起来很奇怪，但它会显着改善你在面试中传达答案的方式，并真正改变我们的体验。扮演候选人和面试官的角色，提问和回答，就像两个人在面试中一样。

3.3 与同行一起练习

自己练习只会带你走这么远。编码面试的主要挑战之一是必须在你做的时候传达你正在做的事情。因此，我们强烈建议与同行面试您一起练习实时编码面试。

如果可能的话，一个很好的起点是和朋友一起练习。如果您的朋友有软件工程师面试的经验，或者至少熟悉该过程，这将特别有用。如果您的网络中没有任何人可以面试您，您可以在我们的[模拟面试平台](#)上找到同行一起练习。

3.4 与前面试官一起练习

与同龄人一起练习的主要好处是它是免费的。但是在某些时候，你会开始注意到你从同龄人那里得到的反馈对你没有那么大的帮助了。一旦你达到那个阶段，我们建议与来自顶级科技公司的前面试官一起练习。

如果你认识一位在 Facebook 或另一家大型科技公司有过面试经验的软件工程师，那就太棒了。但对于我们大多数人来说，很难找到合适的联系来实现这一点。除非你非常了解他们，否则与那个人一起练习几个小时也可能很困难。

www.jiuzhang.com