FAANG Interview. Бортовые заметки сообщества

Last Update: 17/04/2020, [self-link](https://docs.google.com/document/d/1RKzJA7UHj3UKMFxK4Bluy-gB7Sf2fk0mUlCPs76Z07k/edit?usp=sharing)

[Введение](#_x8ekitdu9aeq)

[Аббревиатуры](#_435tiaro6bks)

[Разные полезные ссылки](#_xuskvehwe5nl)

[Курсы, книги, статьи по алгоритмам и структурам данных](#_k0uh8mtdpqao)

[Интересные задачи (списки задач)](#_gpp7h1z6nm76)

[System Design Interview](#_793zqiwslebx)

[Behavioral interview](#_e2wuwyhym9cj)

[Переговоры по офферу](#_ywck0gg7xwk7)

[Составление резюме](#_kb3mwsd621xr)

[Проверка резюме](#_n2yg4aa62uwz)

[SRE](#_4oxcssdpzvxu)

[Интернатура](#_7z0k4lmhtxln)

[Релокация](#_vs55w0kifig)

[Разное](#_5ynzomor2lmf)

[Длинные посты, истории прохождения в ФААНГ](#_8b9yb2lx9fb9)

## Введение

Этот док создан для “FAANG Interview” сообщества с целью упорядочить полезные ресурсы упоминаемые в чате. Каждая ссылка в документе должна сопровождаться описанием. Не стесняемся добавлять/редактировать и делать документ лучше с каждым днем.

[**https://t.me/FaangInterview**](https://t.me/FaangInterview) **- наша телеграм группа**

[**Наш док с рефералами**](https://docs.google.com/spreadsheets/d/1w9W-WmnerMSZryvdYyB7UQLfLJVExuuaoG5lExssnMQ/edit?usp=sharing) **(in progress)**

## Аббревиатуры

* **LP** - Leadership Principles (by Amazon)
* **CtCI** - Cracking the Coding Interview (book)
* **SRE** - Site Reliability Engineer
* **SWE/SE** - Software Engineer
* **SDE** - Software Development Engineer (Amazon)
* **SysDE** - System Development Engineer (Amazon)
* **OA** - Online Assessment
* **HC** - Hiring Committee
* **HM -** Hiring Manager
* **LC** - LeetCode

## Разные полезные ссылки

* Сайты где можно решать задачи:
  + [**https://leetcode.com/**](https://leetcode.com/)
  + <https://www.interviewbit.com/>
* [**Как попасть в Google: инструкция по подготовке**](https://dou.ua/lenta/articles/google-interview/) **- статья Сергея Семы о подготовке к интервью в фаанг.**
* <http://larrr.com/> - блог Ларисы о Гугл, интервью, жизни в долине.
  + [**Хочу работать в Google: Read Me First!**](http://larrr.com/wp-content/uploads/2016/10/InterviewPreparationGuide.pdf) **- подборка материалов по подготовке**
  + <https://t.me/empireinthemaking> - телеграмм канал Ларисы
* [**Cracking the coding interview book**](https://www.amazon.com/Cracking-Coding-Interview-Programming-Questions/dp/0984782850)
* Mock интервью
  + <https://www.pramp.com/#/> - парные мок интервью
  + <https://www.toptal.com/#get-terrific-hackers> здесь можно зарегаться и проверить себя на нескольких интервью (английский, техническое, на дизайн). все бесплатно. если не прошли, можно через какой-то период еще раз попробовать (вроде месяц)
  + <http://www.gainlo.co>/ платный. позволяет собеседоваться с реальными сотрудниками из ФААНГ. сайт гарантирует это. цена указывается своя на каждого интервьюера, от 100 до 200 долларов за мок. обещают подробный фидбек после мока.
* [Довольно большая коллекция книг, собирал из разных источников](https://drive.google.com/open?id=0BzXiuTApZyV5V3ZpeTRKSWgyOTQ)
* Гайды по подготовке к интервью
  + [Interview preparation Guide](https://paper.dropbox.com/doc/Interview-Preparation-Guide-PHeQLxMXugqC1vz72wgtX) - подборка ссылок на полезные репозитории, видео, курсы, и книги.
  + [Ilyushin/google-interview-university: A complete daily plan for studying to become a Google software engineer.](https://github.com/Ilyushin/google-interview-university)
  + [Tech Interview Handbook](https://yangshun.github.io/tech-interview-handbook/)
  + [Coding Interviews: Questions, Analysis & Solutions (Expert's Voice in Programming)](https://www.amazon.com/Coding-Interviews-Questions-Solutions-Programming-ebook/dp/B00ACC6AQY)
  + [Uber Software Developer Interview Prep](https://s3.amazonaws.com/ubercandidateprep/index.html) - подготовка к интервью в Uber
  + [The Interview Study Guide For Software Engineers - DEV Community](https://dev.to/seattledataguy/the-interview-study-guide-for-software-engineers-764)
* [LinkedIn участников чата для обмена контактами](https://docs.google.com/spreadsheets/d/110KLHHLj12T4osSLw2Gl0ttcp2lrQ0RlKeRINBFv6mE/edit#gid=474130129)
* [Статья про интервью в Амазон от одного из менеджеров](https://medium.com/@allo/%D0%BE-%D1%81%D0%BE%D0%B1%D0%B5%D1%81%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B8-%D0%B2-%D0%B0%D0%BC%D0%B0%D0%B7%D0%BE%D0%BD-27e649323c4b)

## Курсы, книги, статьи по алгоритмам и структурам данных

* Курсы
  + [**Курсы по алгоритмам от Роберта Седжвика Часть 1**](https://www.coursera.org/learn/algorithms-part1)[**Курсы по алгоритмам от Роберта Седжвика Часть 2**](https://www.coursera.org/learn/algorithms-part2)
  + <https://cses.fi/book/book.pdf> - Вот хорошая книга для подготовки для интервью. Там и DP есть. Она в основном для тех кто готовится к олимпиадам, т.е. уровень выше, но покрывает и темя которые попадаются в интервью
  + [Coderust: Hacking the Coding Interview](https://www.educative.io/courses/coderust-hacking-the-coding-interview?authorName=Coderust) Платный курс (в текстовом виде) по алгоритмам и структурам данных
  + [Master the Coding Interview: Data Structures + Algorithms](https://www.udemy.com/course/master-the-coding-interview-data-structures-algorithms/) - курс на udemy
  + [JavaScript (JS) Algorithms and Data Structures Masterclass](https://www.udemy.com/js-algorithms-and-data-structures-masterclass/) Курс ознакомляет с основными принципами решения задачек, основными алгоритмами и структурами данных
  + [Coding Interview Bootcamp Algorithms, Data Structures Course](https://www.udemy.com/coding-interview-bootcamp-algorithms-and-data-structure/) На практике дает возможность решить самые популярные задачки, потом идет решение с его разбором с несколькими подходами. Показывает как на практике применять те или иные подходы и оценивать их сложности (big O) и выбирать лучшее решение.
  + [Algorithmic Toolbox](https://www.coursera.org/learn/algorithmic-toolbox?specialization=data-structures-algorithms) Курс для тех кто в начале пути. Поможет понять big-O, элементарные алгоритмы и подходы к ним.
  + [Computer Sc - Data Structures and Algorithms](https://www.youtube.com/playlist?list=PLBF3763AF2E1C572F) видео курс
  + [Algorithms | Computer science | Computing](https://www.khanacademy.org/computing/computer-science/algorithms) - курс по алгоритмам на Khan Academy
  + [Grokking the Coding Interview: Patterns for Coding Questions](https://www.educative.io/courses/grokking-the-coding-interview) платный курс “Patterns for Coding Questions”
  + [Data Structures and Algorithms](https://www.coursera.org/specializations/data-structures-algorithms) - coursera course
  + [Алгоритмы: теория и практика. Методы](https://stepik.org/course/217/syllabus)
  + [Introduction to Algorithms](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/index.htm) - course from MIT
  + [Lectures in Advanced Data Structures (6.851)](https://courses.csail.mit.edu/6.851/fall17/lectures/) - MIT lectures by prof. Erik Demaine
  + [Intro to Algorithms and Data Structures](https://www.udacity.com/course/data-structures-and-algorithms-in-python--ud513) от Google на Python (бесплатно, в конце есть разбор всех этапов онсайт интервью)
* Книги
  + [**Algorithms, 4th Edition by Robert Sedgewick and Kevin Wayne**](https://algs4.cs.princeton.edu/home/) **- более расширенный вариант курсов выше.** [**слайды к книге**](https://algs4.cs.princeton.edu/lectures/)
  + [Computer Science Distilled: Learn the Art of Solving Computational Problems](https://www.amazon.com/Computer-Science-Distilled-Computational-Problems/dp/0997316020)
  + [Algorithms by Jeff Erickson](http://jeffe.cs.illinois.edu/teaching/algorithms/)
  + [Data Structures and Algorithms in Python](https://www.wiley.com/en-ua/Data+Structures+and+Algorithms+in+Python-p-9781118549582) книга по алгоритмам на Python
  + [Problem Solving with Algorithms and Data Structures using Python](https://runestone.academy/runestone/books/published/pythonds/index.html) еще одна книга по алгоритмам на Python
  + [Introduction to Algorithms, 3rd Edition (The MIT Press)](https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844)
  + [Mathematics for Computer Science](https://courses.csail.mit.edu/6.042/spring17/mcs.pdf) - book by Eric Lehman
  + [Структуры данных и алгоритмы в Java](https://www.ozon.ru/context/detail/id/23529814/)
* Разные полезности
  + [Algorithms and Data Structures Cheatsheet](https://algs4.cs.princeton.edu/cheatsheet/)
  + [Algorithms and Complexity questions](https://leetcode.com/discuss/interview-question/469325/google-recruiter-phone-algorithms-and-complexity) неожиданно сплошная теория
* Dynamic programming & Backtracking
  + [Dynamic Programming for Interviews](https://www.byte-by-byte.com/dpbook/) великолепная книга собирающая в себе Top-Down, Bottom Up approaches. Дающая методологию разбора задачи.
  + [Dynamic Programming for Coding Interviews: A Bottom-Up approach to problem solving](https://www.goodreads.com/book/show/34394300-dynamic-programming-for-coding-interviews) книга
  + [Динамическое программирование (видео лекции)](https://www.youtube.com/watch?v=AawQnuYSY4Y&list=PLUfHxBkkFMScK6mOOWp5s6LgbzmtfwmYQ) - курс по динамическому программирования, отлично дает глубокое понимание этого подхода, а также возможность попрактиковаться на решении задач с их разбором
  + [Backtracking introduction (geeksforgeeks)](https://www.geeksforgeeks.org/backtracking-introduction/)
  + [Grokking Dynamic Programming Patterns for Coding Interviews](https://www.educative.io/courses/grokking-dynamic-programming-patterns-for-coding-interviews) - платный курс по динамическом программированию
  + [Grokking Algorithms](https://www.manning.com/books/grokking-algorithms) глава объясняющая KnapSack проблем.
  + [Dynamic Programming lecture #1 - Fibonacci, iteration vs recursion](https://youtu.be/YBSt1jYwVfU) - Errichto дает качественное объяснение приемов Dynamic Programming
  + [Dynamic Programming lecture #2 - Coin change, double counting](https://youtu.be/1mtvm2ubHCY) video
  + [Dynamic Programming Patterns](https://leetcode.com/discuss/general-discussion/458695/dynamic-programming-patterns) карта задач.
  + [Перебор. Жадные алгоритмы. Динамическое программирование](https://foxford.ru/wiki/informatika/odnomernoe-dinamicheskoe-programmirovanie-kolichestvo-sposobov) - видео теория для школьников от Владимира Гуровиц.
* Видео
  + [Introduction to Big O Notation and Time Complexity](https://www.youtube.com/watch?v=D6xkbGLQesk&t=1251s)
  + [Плейлисты про решение задач на вайтборде](https://youtube.com/user/tusharroy2525/playlis)
  + [Видео курс по алгоритмам от автора Cracking the coding interview](https://www.youtube.com/playlist?list=PLI1t_8YX-ApvMthLj56t1Rf-Buio5Y8KL)
  + [Видео курс по структурам данных от автора Cracking the coding interview](https://www.youtube.com/watch?v=IhJGJG-9Dx8&list=PLI1t_8YX-Apv-UiRlnZwqqrRT8D1RhriX)
  + [Sorting Algorithms (slower, grouped and ordered)](https://www.youtube.com/playlist?list=PLZh3kxyHrVp_AcOanN_jpuQbcMVdXbqei) - анимация 26 сортировок ([нечто похожее](http://sorting.at/))
  + [Back To Back SWE](https://www.youtube.com/channel/UCmJz2DV1a3yfgrR7GqRtUUA) понравился этот канал. У парня хороший английский, он хорошо объясняет подходы к решению разных задач с описанием процесса решения, появления инсайтов
  + [WilliamFiset](https://www.youtube.com/channel/UCD8yeTczadqdARzQUp29PJw) - серия видео о теории графов
  + Списки видео: [базовые алгоритмы](https://www.youtube.com/playlist?list=PLrS21S1jm43geDXVdeQy96P-f59pXeyPC) и [чуть сложнее базовых](https://www.youtube.com/playlist?list=PLrS21S1jm43iF3DKP3rvpN8hoTBqHVYbr)

## Интересные задачи (списки задач)

* Общие принципы
  + [Очередность Leetcode задач для новоприбывших.(чтобы меньше страдать)](https://docs.google.com/spreadsheets/d/1je6J87BX5C5fo5Gbok1TJncVK3-UFkiDznHUzhtHbVU/edit#gid=0) и ([объяснение по какому принципу они фильтровались](https://leetcode.com/discuss/general-discussion/522705/1000-leetcode-problems-within-a-year))
  + [A LeetCode Grinding Guide - cscareerquestions](https://www.reddit.com/r/cscareerquestions/comments/6luszf/a_leetcode_grinding_guide/) - обсуждение как правильно начинать решать задачи на литкоде.
  + [14 шаблонов, которые помогут ответить на любой вопрос по коду на собеседовании Вопросы на собеседованиях часто](https://tproger.ru/translations/14-templates-to-answer-interview-questions/) (это “переписка” курса Grokking the Coding Interview: Patterns for Coding Questions который уже указан выше)
* Отдельные задачи
  + [Palindromic Substrings](https://leetcode.com/problems/palindromic-substrings/description/) - задача с колл в Uber
  + [Разбор задачи с собеседования в Google: синонимичные запросы](https://m.habr.com/ru/post/437702/)
  + [Решаем задачу из интервью Google на JavaScript: 4 разных способа](https://m.habr.com/ru/company/skillbox/blog/443886/)
  + [Google Interview Problems: Ratio Finder](https://medium.com/@alexgolec/google-interview-problems-ratio-finder-d7aa8bf201e3)
* Подборки задач
  + <https://www.interviewbit.com/courses/programming/> - подборка задач по темам от interviewbit
  + [List of Top 75 LeetCode Questions to Save Your Time](https://www.teamblind.com/article/New-Year-Gift---Curated-List-of-Top-100-LeetCode-Questions-to-Save-Your-Time-OaM1orEU)
  + [Amazon Online Assessment Questions](https://leetcode.com/discuss/interview-question/344650/Amazon-Online-Assessment-Questions) из них было Critical Routers, и Product Suggestions
  + [Amazon Final Interview Questions | SDE1](https://leetcode.com/discuss/interview-question/488887/amazon-final-interview-questions-sde1)
  + [Leetcode questions by likes/dislikes ratio](https://docs.google.com/spreadsheets/d/1LXSO4_Kz_1JMqoZMMuvhpBI99kvsZA4vSSCRAznqnXM/edit?usp=sharing)
* Видео
  + [Binary Search with Duplicates video](https://www.youtube.com/watch?v=vHv8AiWf0M4)
  + [Mock Google interview (for Software Engineer job) - coding & algorithms tips](https://www.youtube.com/watch?v=IWvbPIYQPFM)
* [https://geeksforgeeks.org](about:blank) - разбор множества задач и алгоритмов
  + [Anagram Substring Search (Or Search for all permutations)](https://www.geeksforgeeks.org/anagram-substring-search-search-permutations/)
  + [Construct tree from ancestor matrix](https://www.geeksforgeeks.org/construct-tree-from-ancestor-matrix/)
  + [Top 10 algorithms in Interview Questions](https://www.google.com/amp/s/www.geeksforgeeks.org/top-10-algorithms-in-interview-questions/amp/)
  + [Bit Tricks for Competitive Programming](https://www.geeksforgeeks.org/bit-tricks-competitive-programming/)

## 

## System Design Interview

* [**Grokking the System Design Interview**](https://www.educative.io/courses/grokking-the-system-design-interview) **в 99% в первую очередь вам посоветуют пройти курс (**[Grokking the System Design Interview - Part 3](https://coursehunters.online/t/educative-io-design-gurus-grokking-the-system-design-interview-part-3/581) **)**
* [Grokking the Object Oriented Design Interview](https://www.educative.io/courses/grokking-the-object-oriented-design-interview)
* [System-design-primer](https://github.com/donnemartin/system-design-primer) Популярный сборник разного по SDI
* [System Design Interview Step By Step Guide](https://www.youtube.com/channel/UC9vLsnF6QPYuH51njmIooCQ/videos). Youtube channel
* <https://dataintensive.net/> одна из самых рекомендуемых и полезных книжек по этой теме
* [Database Internals: A Deep Dive into How Distributed Data Systems Work](https://www.amazon.com/Database-Internals-Deep-Distributed-Systems/dp/1492040347) - книга говорят бомба типа data intensive
* <https://landing.google.com/sre/books/> не пугайтесь что про SRE, для SWE можно кой чего пропустить - но есть очень много примеров дизайнов, решений которые были приняты и почему, очень очень хорошее описание что такое load balancing например
* [Introduction to modern network load balancing and proxying](https://blog.envoyproxy.io/introduction-to-modern-network-load-balancing-and-proxying-a57f6ff80236) - Еще про load balancer, но это уже чуть подробней чем надо SWE обычно,
* [System Design Interview Video. Rate Limiting (local and distributed)](https://youtu.be/FU4WlwfS3G0?list=LLVmH0b-Ae4v-ow9l2f5c5bw)
* [Prepare for Your Google Interview: Systems Design](https://www.youtube.com/watch?v=Gg318hR5JY0) неожиданно крутое новое видео по system design от Life at Google
* [Designing Data-Intensive Applications (DDIA) — an O'Reilly book by Martin Kleppmann (The Wild Boar Book)](https://dataintensive.net/) book
* [Object-Oriented Design](https://leetcode.com/discuss/interview-question/object-oriented-design?currentPage=1&orderBy=hot&query=) - список вопросов с литкода
* Старые статьи с DOU о консенсус-протоколах: [раз](https://dou.ua/lenta/articles/12-konsensus-protocols), [два](https://dou.ua/lenta/articles/konsensus-protocols-2/)
* [Raft](http://thesecretlivesofdata.com/raft) Объяснение RAFT на пальцах с красивыми диаграммками. Можете даже показать свой маме!
* [Grokking the Mobile System Design interview](https://medium.com/@goncharov.artemv/grokking-the-mobile-system-design-interview-6a06fa94491b?source=friends_link&sk=5d35ee6ade621e3a8018e048d25f6009) - По мотивам систем дизайн собесов в ФБ и Гугл на мобильщика написал статейку.
* Видео
  + [Software Design Patterns and Principles (quick overview)](https://www.youtube.com/watch?v=WV2Ed1QTst8)
  + [Success in Tech](https://www.youtube.com/channel/UC-vYrOAmtrx9sBzJAf3x_xw/featured) - канал о подготовке к интервью, есть видео о дизайне
  + [System Design Interview](https://www.youtube.com/channel/UC9vLsnF6QPYuH51njmIooCQ) канал по дизайн интервью
  + лучше/популярней и понятней лекций по Raft/Paxos я не видел, а глубже наверно только статьи читать: [Raft lecture](https://www.youtube.com/watch?v=YbZ3zDzDnrw), [Paxos lecture](https://www.youtube.com/watch?v=JEpsBg0AO6o)

## Behavioral interview

* Видео которое рекомендуют везде: [Episode 07: Intro to Behavioural Interviews](https://www.youtube.com/watch?v=PJKYqLP6MRE)
* Тематический канал [Дена Кройтора](https://www.youtube.com/channel/UCw0uQHve23oMWgQcTTpgQsQ/playlists), много инфы и примеров по Amazon LP, FB, Netflix
* [Behavioral Interview. Топики на эту тему на блаинде](https://www.teamblind.com/issue/13246/Behavioral-Interview)
* Cracking the Coding Interview pages 32-38
* [The Situation-Behavior-Impact-Feedback Tool - From MindTools.com](https://www.mindtools.com/pages/article/situation-behavior-impact-feedback.htm)
* <https://leetcode.com/explore/interview/card/leapai/>
* Блог bar-raiser из Amazon о их LP: [Interviewing at Amazon — Leadership Principles - Dave Anderson](https://medium.com/@scarletinked/are-you-the-leader-were-looking-for-interviewing-at-amazon-8301d787815d)
* [Amazon Behavioral questions | Leadership Principles | LP](https://leetcode.com/discuss/interview-question/437082/amazon-behavioral-questions-leadership-principles-lp) Тут ещё один товарищ из этого чата поучаствовал в эвенте, и у нас у обоих было одинаково: всё вопросы на LP & Behavior были слово в слово из этого списка
* Если это **нетфликс или амазон** там нужно читать их пейперы по кор вельюсам и быть готовы к вопросам показывающим как ты им соответствуешь.

## Переговоры по офферу

* [How to Negotiate Your Job Offer - Prof. Deepak Malhotra (Harvard Business School) Video](https://www.youtube.com/watch?v=km2Hd_xgo9Q)
* [Ten Rules for Negotiating a Job Offer](https://haseebq.com/my-ten-rules-for-negotiating-a-job-offer/)
* [Patrick McKenzie | Salary Negotiation: Make More Money, Be More Valued](https://www.kalzumeus.com/2012/01/23/salary-negotiation/)
* [Salary Negotiation with Haseeb Qureshi](https://softwareengineeringdaily.com/2016/07/11/salary-negotiation-with-haseeb-qureshi/)
* [How Not to Bomb Your Offer Negotiation](https://haseebq.com/how-not-to-bomb-your-offer-negotiation/)
* [How I negotiated a $300,000 offer in Silicon Valley](https://medium.com/@bayareabelletrist/how-i-negotiated-a-software-engineer-offer-in-silicon-valley-f11590f5c656)

## Составление резюме

* От Ларисы
  + [Пример хорошего резюме](http://larrr.com/primer-horoshego-rezyume/)
  + [resume Archives](http://larrr.com/category/resume/) - раздел о резюме в блоге Ларисы.
  + [Resume Checklist или Как Делать Не Надо](http://larrr.com/resume-checklist-ili-kak-delat-ne-nado/)
  + [Список ключевых слов-достижений](http://larrr.com/hochu-rabotat-v-google-klyuchevye-slova-pri-sostavlenii-rezyume/)
* Видео
  + [Советы от гугл рекрутеров](https://youtu.be/BYUy1yvjHxE)
  + [7 Tips for the Coding Resume (for Software Engineers)](https://www.youtube.com/watch?v=xpaz7nrNmXA) video

## Проверка резюме

* По вторникам и субботам в [/r/cscareequestions](https://www.reddit.com/r/cscareerquestions/) создаются “Resume Advice” тредs, где можно получить фидбек.
* [Rooftop Slushie](https://www.rooftopslushie.com/topics/resume-review)

## SRE

* [Linux Technical Interview Questions and Answers](https://www.udemy.com/course/linux-technical-interview-questions-and-answers/) (Linux) - платный курс
* [My Job Interview at Google](https://catonmat.net/my-job-interview-at-google) Google SRE interview on SRE
* [What happens when we hit URL in Browser?](https://www.freecodecamp.org/news/what-happens-when-you-hit-url-in-your-browser/)
* [SYSADMIN. The Systems Engineering Side of Site Reliability Engineering](https://www.usenix.org/system/files/login/articles/login_june_08_hixson.pdf)
* [SYSADMIN. Hiring Site Reliability Engineers](https://www.usenix.org/system/files/login/articles/login_june_07_jones.pdf)
* Книги
  + [Site Reliability Engineering](https://landing.google.com/sre/books/) книга
  + [Linux System Programming: Talking Directly To The Kernel And C Library](https://www.amazon.com/Linux-System-Programming-Talking-Directly/dp/1449339530/ref=sr_1_12?crid=2BXSSOER0QTDS&keywords=linux+programming&qid=1579717475&sprefix=linux%2Caps%2C234&sr=8-12)
  + [Systems Performance: Enterprise and the Cloud](https://www.amazon.com/Systems-Performance-Enterprise-Brendan-Gregg/dp/0133390098/ref=sr_1_2?crid=29XENXAQ8F4U6&keywords=brendan+gregg&qid=1579717501&sprefix=brandan+gre%2Caps%2C243&sr=8-2), сайт автора (<http://www.brendangregg.com>)
  + [BPF Performance Tools (Addison-Wesley Professional Computing Series)](https://www.amazon.com/Performance-Tools-Addison-Wesley-Professional-Computing/dp/0136554822/ref=sr_1_1?crid=29XENXAQ8F4U6&keywords=brendan+gregg&qid=1579717501&sprefix=brandan+gre%2Caps%2C243&sr=8-1) снова (<http://www.brendangregg.com>)
  + [Operating Systems: Three Easy Pieces](http://pages.cs.wisc.edu/~remzi/OSTEP/)
  + [The Linux Programming Interface: A Linux and UNIX System Programming Handbook](https://www.amazon.com/Linux-Programming-Interface-System-Handbook/dp/1593272200/ref=sr_1_1?crid=2BXSSOER0QTDS&keywords=linux+programming&qid=1579717475&sprefix=linux%2Caps%2C234&sr=8-1)

## Интернатура

* [Summer 2020 Internships](https://github.com/elaine-zheng/summer2020internships)
* [Статьи о стажировке в фаанг](https://dou.ua/lenta/tags/%D0%BF%D1%83%D1%82%D1%8C%20%D1%81%D1%82%D0%B0%D0%B6%D0%B5%D1%80%D0%B0/)

## Релокация

* [Релокация в Сингапур: история украинского архитектора](https://dou.ua/lenta/articles/relocation-to-singapore/) Хорошая статья о жизни в Сингапуре, финансах и т.д.
* [Как попасть в Кремниевую Долину. Video](https://youtu.be/bsrW1qQGKcQ)
* <https://t.me/FbCovid> - мини группа для обсуждения переезда в ФБ Лондон в текущей ситуации с вирусом. Для тех, кто сейчас находится в процессе обсуждения офера или уже подписал его, но еще не переехал из-за вируса.

## Разное

* Истории из жизни:
  + [Why I studied full-time for 8 months for a Google interview](https://www.freecodecamp.org/news/why-i-studied-full-time-for-8-months-for-a-google-interview-cc662ce9bb13/) - Статья про подготовку в Google за 8 месяцев ( ) +[русская версия](https://proglib.io/p/8-month-for-google-interview/)
  + [Про то, как я попал на интервью в Microsoft Ireland](https://medium.com/@ykurmangaliyev/%D0%BF%D1%80%D0%BE-%D1%82%D0%BE-%D0%BA%D0%B0%D0%BA-%D1%8F-%D0%BF%D0%BE%D0%BF%D0%B0%D0%BB-%D0%BD%D0%B0-%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%B2%D1%8C%D1%8E-%D0%B2-microsoft-ireland-86a3101b335a)
  + [Подготовка к собеседованию в Microsoft: polycode](https://polycode.livejournal.com/29426.html)
  + [Как подготовиться к собеседованию в Google и не пройти его. Дважды](https://habr.com/ru/post/419945/)
* [Комбинаторика: основные правила и формулы.](https://ya-znau.ru/znaniya/zn/80)
* [Вся правда о собеседованиях в Google: за пределами NDA](http://blogerator.org/page/google)
* [amazon Banned Questions](https://docs.google.com/spreadsheets/d/1bJRbxssWmP5C8iv5MQvOkh4k2LeCgLsjbaosrIBVPf0/edit#gid=0)
* [Amazon 1st round Interview questions (manager level)](https://www.reddit.com/r/ITCareerQuestions/comments/cz9ctv/amazon_1st_round_interview_questions_manager_level/?utm_medium=android_app&utm_source=share)
* [Levels.fyi: Compare salaries and career levels across companies](https://www.levels.fyi/)
* Книги
  + [Introduction to Information Retrieval](https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf) (на русском [Вильямс книга Введение в информационный поиск](http://www.williamspublishing.com/Books/978-5-8459-1623-5.html) )
* Сайты для изучения программирования
  + [Interview Practice](https://app.codesignal.com/interview-practice) - средние между leetcode и codewars
  + <https://www.codewars.com> - для изучения языков, обилие задач и технических приемов
  + [Firecode.io | Coding Interview Answers and Trainer](https://www.firecode.io/)
* Список ресурсов, на которых можно найти рефералов (из какой-то статьи):
  + <https://repher.me/>
  + <https://www.teamblind.com>
  + <https://www.rooftopslushie.com/> - за деньги
  + <https://triplebyte.com/>
* Списки компаний по регионам
  + [BigN-style Interview Europe](https://docs.google.com/spreadsheets/d/1jJehhHdt1dHHnCtbWqLjc4o-GKBdlDixgma7oJXlBMQ/edit#gid=0)
  + [Europe companies](https://docs.google.com/spreadsheets/d/1HTmdCYbdF22n3irlvvdfmfAY7WZDR_-pwlO8FJrYHlY/edit#gid=0)

## Длинные интересные посты ("#пробовался\_в\_фаанг и полезные мысли )

**@matveevtut:** Очень плюсую Седжвика, в своё время помог мне устроиться в яндекс, а там собеседования по сути мало чем отличаются от FAANG, так же 5 часов тебя мучают. Может только конкуренция поменьше.

Короче, после окончания двух частей курса Седжвика на курсере с прорешиванием всех домашек я там все задачки просто в клочки разорвал.

Ещё в яндексе была отдельная часовая секция по теории алгоритмов, без написания кода. Там сначала опрашивали просто по теории, как на экзамене, а потом просили придумать алгоритм для той или иной задачи и разобрать его сложность. Так вот, в этой секции Седжвик тоже очень вывез.

А закончилась теоретическая секция вопросом по NP-полной задаче и придумыванием эвристик, её упрощающих. Вот это я не вывез.

Ещё могу посоветовать очень хороший курс Тима Рафгардена (Tim Roughgarden), это как бы конкурент Седжвика на курсере. Я теперь перед FAANG его прохожу, потому что у него много глубокого материала, который Седжвик вообще не освещает, как бы алгоритмы с другой стороны: рандомные алгоритмы, их анализ с применением теорвера, NP-полные задачи и эвристики.

**@john\_krohl** я уже второй год подписан на Leetcode. Премиум открывает все задачи, открывает сортировку алгоритмов по компании, открывает детальную инфу по каждому алгоритму (в какой компании спрашивался, и тд.) и еще кучу всего чем я пользуюсь редко. Можно ли обойтись без подписки? - ДА, она не обязательна. Удобнее ли с ней - Да, с ней намного удобнее. Влияет ли она на качество подготовки - Нет, все зависит только от тебя. Есть ли смысл платить премиум? - Все зависит от твоего дохода, если 35$ это для тебя большие деньги, то не стоит, вся инфа на leetcode легко гуглится за 5 сек. Если же 35$ это то сколько ты платишь за поход в кино или в ресторан, то на мой взгляд лучше один раз туда не пойти и купить подписку, пользы будет больше

**@sparfenyuk** живу и работаю в мюнхене, но в гугл особо не торопился. однако, на меня вышел рекрутер с предложением прособеседоваться на роль С++ инженера. отказываться было глупо, но отсутствие предварительной подготовки (4-6 месяцев? пфф) могло сказаться на моем успехе (спойлер - так и вышло). первое общение было с тем же рекрутером на общие темы: мой питч о себе об успехах в карьере, ее рассказ о проекте и в конце краткий блиц по вопросам CS. этот скрининг я прошел и оказался на техническом интервью с инженером гугла. попался русскоязычный парень, но общались мы все так же на английском. на этом собесе решали задачки. точно не вспомню, какие но уровня медиум, две штуки. между нами был пошарен гугл док, где он кратко описал условия задачи, а я - свой код. все бы ничего, но связь была с аудиопомехами, из-за чего было сложно иногда понимать собеседника. по итогам этого этапа предложили пройти следующее интервью уже с другим инженером. опять задачи были относительно несложными, но я чутка закопался в коде, из-за чего вторую задачу мы обсудили только устно. к сожалению, дальше я не прошел. но и такой опыт - тоже опыт.

мои выводы для себя:

- поддерживать свои навыки в актуальном состоянии, даже если не в активном поиске

- делать акцент на прорешивании вслух, а лучше на сервисах вроде pramp

**@john\_krohl** заранее напишу, что контракта у меня нет.

Живу и работаю в Мюнхене. Лет 5 назад на почту написал рекрутер из Facebook. Предложил пройти телефонное интервью. Сказал «повтори там алгоритмы, сортировку, обход дерева и тд» я за неделю повторил все и завалил интервью. Мне сказали, чтобы я не расстраивался и в следующем году повторил еще раз. В следующем году мне никто не написал, а рекрутер не отвечал на письма. Тогда же я решил, что в любом случае нужно заниматься алгоритмами, но не так чтобы «повтори сортировку», а детально изучить весь аспект проблемы. Начал я с книги “Algorithms, 4th Edition by Robert Sedgewick”. Здоровенная такая книга, прочитал примерно 30% и понял, что книга сложная, и как-то чтение это не мое, решил решать задачи на Leetcode. Взял Easy задачку и не смог ее решить. Почитал объяснение и все равно ничего не понял. То есть сам код я понимаю, там циклы разные и рекурсия, но каким образом оно решает саму проблему не ясно. А как они вообще до этого додумались была чистая магия. Решил, что просто так решать задачки — это глупо, нужно все-таки сами алгоритмы выучить сначала, подписался на coursera на курс “Data Structures and Algorithms Specialization”. Этот курс оказался супер большим и невероятно сложным. Там правда были легкие темы, но домашки полностью выносили мозг. В общей сложности я потратил около 7ми месяцев чтобы полностью закончить этот курс. После этого я дочитал книгу Sedgewick и начал заниматься Leetcode. В какой-то момент мне опять написал Facebook, я с легкостью прошел телефонное интервью, меня пригласили на онсайт. Задачки на онсайт тоже были не сложными, но system design я полностью провалил. Через 6 месяцев Facebook написал еще раз и предложил срочно пройти собеседование на горячую позицию в Лондон. Я уже начал паковать чемоданы, потому что был 100% уверен, что пройду и провалил первое телефонное интервью, не ответив на элементарную задачку которую раньше видел, но забыл что и как. После этого я начал каждый день решать хотя бы по 1й задачке + повторять пройденные, так как они быстро забываются без практики. В какой-то момент мне написал Microsoft, предложили позицию в Праге, я прошел 3 интервью, 2 с задачами и system design. Они сказали что все супер, но позиция требует более высокого уровня system design. Сказали что напишут если будет подходящая позиция, до сих пор пишут. Написали опять из Facebook, спросили хочу ли я пройти интервью? Назначили в сентябре. Между этим написали в LinkedIn из Google и спросили хочу ли я пройти собеседование к ним. Я подумал что было бы здорово попрактиковаться в Гугл перед собеседованием в Фейсбук. Я в принципе уверенный в алгоритмах, но system design может подкачать, поэтому готовился именно к этому изучая все что мог найти. В итоге на phone screening я получил Hard задачу которую не смог решить =) Интервьюер сказал что эту задачу никто не может решить за 20 минут, и то что я не расплакался и не убежал уже его радует и пожелал мне удачи в дальнейших интервью. Потом было стандартных 5 интервью в офисе гугл. Было очень сложно. Я решил задачи, но 1 решил не самым оптимальным способом. В итоге рекрутер сказал, что вроде как норм, будут искать команду и только когда найдут, сделают офер. До сих пор ищут команду, открытых позиций буквально 2-3 и я на них не подхожу. Я пишу на Java, вакансии на С++. Я по-прежнему готовлюсь к собеседованию в Facebook и решаю Leetcode каждый день и параллельно читая CCI.

мои выводы для себя:

- (я это скопировал из предыдущего поста, полностью согласен) поддерживать свои навыки в актуальном состоянии, даже если не в активном поиске.

- Заваленное интервью это знак, что нужно готовиться сильнее.

- Лучше завалить 10 интервью, чем ничего не делать.

**@kotsabiukmv** Не знаю как надо отвечать. Я отвечал следующим образом:

"Tell me about your recently implemented task." рассказал о своей одной из последних тасок которую считал наиболее подходящей, немного в красках и с восторгом

" Imagine you work with someone in a team with who it is really hard to talk. How would you collaborate with him and try to get consensus for some task?" Сказал что говорил бы с этим человеком через кого кто для нее авторитет, или выносил бы эти вопросы на митинг со всей командой

" Imagine you work in a company with a bad culture, what would you do in this situation?" сказал что не молчал бы а рассказывал бы начальству о том какая культура в компании, может он об этом даже не знает, если бы меня не слушали то присоединился поддержкой людей, которые разделывают мои взгляды

"Imagine you are a director of a company with a bad culture, what would you do in this situation?" Сказал что говорил бы с подчиненными чтобы узнать что им не нравится, старался бы решить эти проблемы, трансформировать компанию както, если бы не получалось, то приобщился бы помощью специалистов - соответствующих консультантов

" Describe how organized development process in your team." описал процесс как есть, только в красках и с эпитетами

**@svolodko** Привет. Что ж, я отстрелялся. У меня был один вопрос на ООП, типа реализовать утилиту, которая ищет файлы по имени директории + списку необязательных входных параметров. Типа find в Unix. Это был кодинг, но не про задачки,а про что-то более базовое и часто встречающееся.

Второй и третий вопросы были классический LeetCode: один почти слово в слово Knapsack problem, второй - голый Topological Sort в направленном графе.

Эти три я ответил на хорошо либо даже отлично.

Последний был с Дев менеджером, на System Design, и там всё было плохо. Сначала на LP я бесконечно рассказывал, как я налажал тут, сорвал сроки там, и тд. А потом он попросил задизайнить мне Autocompletion functionality на той же Google search. Я блин так и не прочитал заранее про этот кейс (хотя он у меня был буквально через один в списке не прочтение), я понял, что мне просто надо крутиться, как уж на сковородке, весь этот час, о есть полчаса. Потому что я как раз в SD очень слаб. Так что я продвинулся, как смог. Сейчас смотрю правильный ответ на этот вопрос: я многое сделал правильно, многое, сказанное своими словами, совпадает с тем, как положено отвечать. Но какие-то вещи типа шардирования и кэширования - я так и не мог предложить, по какому принципу хранить тот же финальный Trie , по которому статистика частот считается. То есть на низком уровне я предложил верную дата структуру, но когда речь зашла о больших объёмах, я поплыл. На доске нормальной схемы так и не получилось.

Резюме: систем дизайн завалил, с кодингом справился хорошо. Если команда скажет, что кодит хорошо, а остальному быстро научится, то будет мне оффер. Если же дизайн важен, то получу отказ. Ну и по BI я говорил довольно честно, и они могут решить, что чувак не соответствует их принципам. Это я вообще не уловил, подхожу я им или нет

На меня вышли через ЛинкедИн, притом что у меня профайл почти пустой пока. Но вышли видимо потому, что я территориально в Польше нахожусь, и рекрутеры просеивают локальный рынок. Насколько я знаю, следующий раунд Амазона будет в Лондоне, потом Хорватия, Португалия. Всё вроде как с разницей в месяц. И ещё важно, что рекрутер отдельно просит подтягивать своих друзей и коллег, которым такой эвент был бы интересен. Так что достаточно аукнуть рекрутера - вас включат в первоначальные списки. Правда, перед онсайтом попросят онлайн-ассесмент пройти на две задачки в полтора часа.

Платформа у них собственная. Присылают ссылку и просят пройти по ней в течение недели. Как только прошёл - включается таймер на 90 минут, который никак не остановить. После 90 минут таймер перезапустят еще на 15 минут, и в это время нужно словами (письменно) описать оба решения, какова сложность и почему. Из удобного - вместе с первой ссылкой пришлют вторую, на демо-вход. Таким образом можно заранее освоиться с интерфейсом, что реально помогает, во время самого ассесмента голова не занята поиском нужной функциональности в интерфейсе и тд. Хотя всё равно я умудрился после написания кода первой задачки нажать кнопку Submit вместо Compile ))

**@Chelioz23**

Помнится, я тут обещал написать впечатления от опыта собеседования в Дублинский AWS. Ничего о неразглашении я не подписывал, всё самое интересное уже есть на глаздоре, просто жадно.

Офер не получил, кулдаун 1 год. Вполне справедливое решение, о косяках позже.

Зато понял правила игры. И это был очаровательный месяц с кучей приключений.

Вводные: Network Development Engineer. С референсом. 9 лет опыта, 5 из них у вендора на хорошей позиции, веду проект.

Стандартный процесс: 1 технический on-call, 1 день onsite состоящий из 5 лупов и обеда. Без звонка HR.

Не все девелоперы понимают, что вообще делают сетевые инженеры. Нет, не кабели втыкаем, я коробок своих не видел 2 года. Это не тоже самое, что админить серваки. Есть миллион протоколов и стандартов, и особенности эксплуатации сетей, из-за чего тонна легаси вещей тянет индустрию назад.

Последние лет 5 ещё и просят код писать. Крутую штуку со скейлом сетевому создавать не надо, но репортинг и провиженинг – нужны каждому спецу. Python наше всё.

# Часть 1

Изначально меня зареферили в Австралию, из-за недопонимания. Но по семейным обстоятельствам я туда пока поехать не могу. И первый технический звонок был с Австралией.

Для выбора даты и времени прислали ссылку с календарями, а также опросник в догонку, по мелочи. Им ничего из моих дат не подошло, хотя я пытался забукать в рабочее австралийское время. Пикнули на 7:30. Боль.

Интервьюера было два, один «тень». Из Китая и Индии, говорят нормально, понятнее нейтив австралийцев. Судя по всему, меня опрашивал новичёк. Программа всем известная:

Чуть-чуть про бэкграунд и проекты со скейлом. Желательно с автоматизацией.

Routing protocol на мой выбор. Чел не смог сам в OSPF, так что захватили BGP, но это не по правилам. Очень простые вопросы уровня CCNP.

Как грепнуть что-то в линуксе, умею ли я в питон. Очень поверхностно.

Leadership principles. Расскажи историю, когда ты повёл себя адекватным образом, <деепричастие, описывающее один принцип>

Что не понравилось. Как специалист, по сетевой части – я 100% опытнее и дотошнее. У меня был ужасный опыт работы после такой халявы, но данные опасения рассеиваются онсайтом.

Пока я уточнял у собеседника вопрос - отвечал на следующие 5, но он не стеснялся их повторять. Боялся уйти от плана, не давал заработать экстрапоинты.

Как оценивать, если тебе назвали не 3 ожидаемых способа решения, а 5? При том, кандидат вежливо намекает, что вы ожидаете костыль из книжки, а не нормальное решение.

Это обманчивая простота. Просто такой инженер попался и у него всего 1 час на все темы. На онсайте всё глубже.

# Часть 2

После 1 сессии позвонили и позвали на онсайт. Но выяснилось, что мы говорим про разные страны. Новый технический прогон не понадобился, процесс стандартный и я прошёл унифицированный фильтр на эту роль.

Мало того, процесс стандартный во все отделы, где нужны NDE. Вас будут тянуть в самый нуждающийся, но говорят, что выбор таки есть.

После того HR две недели не присылала мне рекомендательное письмо для визы в Ирландию. Причём 3 раза просила мои паспортные данные, которые я каждый раз писал заново, с reply и указанием на то, что это повтор. Это самое стыдное, что позволил себе Амазон за всё время.

Виза делается за свой счёт. Очень важно: в анкете в причинах поездки правильный ответ “Other”, пояснением, что это interview. Ни туристическая и ни бизнес виза вам нужна. Уникальность всей ситуации в том, что компания обязуется покрыть все расходы, и от вас требуется меньше документов. Никаких билетов и отелей бронировать до получения этого типа визы не надо.

Делали 10 дней, до рождества. Дали на 30 дней, вместо запрошенных 5.

После получения визы вы согласуете дату. Вам высылают список интервьюеров. Вас направляют в тревел агенство, они же букают отель и билеты. В теории можно договориться и на неделю пребывания, просто они не оплатят отель на все дни. У меня было всего 2 дня, из-за болеющей супруги. Инструкций накидали, тупить негде.

Это праздники, возвращался 8го января, билетов нормальных нет. Летел люфтами с большой пересадкой. Отель прямо рядом с офисом, Clayton, с завтраком. Не жлобились, уже на этом этапе потратили на меня ~800 евро. Вы можете зарепортить любые reasonable расходы, типа ужина после интервью и транспорта в аэропорт.

# Часть 3

Onsite. 5 лупов по часу, 45 минут на каждую беседу, но кроме обеда я не отдыхал больше 1 минуты. Этот процесс написан кровью(потраченными деньгами), как пдд, и у каждого из собеседников есть своя роль по теме.

Обед – самое простое. Ланч-бади ведёт вас поесть в любое место за счёт компании. Этот этап не должен оцениваться, и напротив – даёт кандидату оценить компанию положительно. Халява и пофлудить. Любой из собеседников может даже булочку вам купить, и запросить компенсацию у компании.

Набор такой: Бекграунд и софт-скилы, Network Fundamentals, Routing, Automation, Design&HW/Tshoot.

Базовые вещи типа TCP – крайне желательно знать. Все пишут в резюме, мало кто знает TCP в деталях – а зря. Подготовиться можно, желательно понимая характер вопросов.

Это игра с правилами, может не иметь ничего общего с вашей проф компетенцией. Проверка на то, как высоко вы можете прыгнуть. Своеобразный экзамен. Если повезёт, то собеседник будет держать курс на реальные, хоть и вечные, проблемы.

Каждому из интервьюеров даётся один из 14 принципов Амазона, о применении которых должен рассказать кандидат. Просто заготовьте масштабные истории, где есть польза конкретно от вас, до 10 минут. Надо использовать star approach.

Все без исключения хотят и стараются тебе понравиться. Крайне трудно понять, что ты ответил что-то неправильно. Все русские парни – отлично готовились, и являются хорошими специалистами. Реально хреновые вопросы были только на одном лупе, с индусом. Собеседования не записываются, но с русскими надо говорить по-английски. Это неэффективно и тянет время, поэтому вы можете договориться втихую. Тока тссс.

Офера я не получил, из-за технической части. Я самовлюблённый, но переживал всего 2 часа.

Мне позвонили и сказали об этом голосом, приятно. Причём дозвонились только на второй день, но слать отбивки – не по правилам.

Я Customer faced и занимаюсь английским последние 1.5 года, поэтому очень горд за два лупа с менеджерами. Поднять эти вещи может быть в 5 раз дольше и сложнее, чем мои провалы по технике. Я был крайне воодушевлён после каждой беседы, кроме последней.

Готовился, в самолёте, в силу семейных обстоятельств. Немного ещё, пока гулял с ребёнком в коляске. Вопросы были качественно другого уровня, нежели на звонке, но есть ответы, за которые мне стыдно:

1) Слабо понял задание по выбору железа в DC фабрике и траблшутингу. Возможно это задание на рассуждения, но что там рассуждать, если все знают, что у них single chip коробки в DC. Поговорил о начинках железа, которое мне знакомо в вендоре.

Мне попался чувак из Индии, который не умеет формулировать вопросы и абстрагироваться. Его поправлял напарник, но спустя недели я понимаю, что мне надо было не защищаться, а нападать. У коллег к этому товарищу есть претензии, но его бэкграунд вполне внушает.

Буквально затупил про mac-коллизию когда про САМ говорили. Сам же про хеши упомянул, но индус больно агрессивно прыгал по вопросам, чтобы осознать контекст.

Дебильный спич про траблшутинг в том же лупе. По сути, любой tshoot это дерево с вероятностями. Всё что там можно узнать – насколько масштабно я мыслю в пределах Ops.

2) Забыл про особенности кластеризации RR в задачке по BGP. Лайфхак: OSPF и ISIS в трендах мертвы, к ним собеседник готов всегда хуже. Выбор протокола для беседы – за вами.

3) Самый стыдный фейл: не смог внятно объяснить отличия list и tuple. С парсингом и моделью данных я вроде справился, не знаю проверял ли он правильность скобочек после беседы. Также возможно ожидали что-то более детальное про выполнение команды ls .

**@EvgenyPanov**

Предыстория:

Подавался в Амазон через рефер от своего друга, выбрал Systems Development Engineer, Amazon, Ванкувер.

Через пару недель через почту со мной вышел на связь HR, который назначил время первого технического телефона. На первом телефоне было 10 минут Leadership Principles, затем systems/linux и после этого задача на кодинг. Через 3 дня пришло новое письмо от HR, где говорилось, что все ок, давайте проводить второй. На втором телефоне все аналогично первому в плане структуры интервью. На каждом из двух интервью у меня оставалось порядка 15-20 минут, для моих вопросов. Прошло еще 3-4 дня и мне написали, что приглашают меня на онсайт в Ванкувер. Запланировали дату, я начал заниматься визой. Для такого рода поездки, мне было нужно письменное приглашение от Амазон для оформления визы. В итоге за месяц они так и не смогли прислать приглашение, я уже даже не надеялся поехать. Но в очередной раз, когда я пинговал HR'ов, поставил в копию еще одного, с которым взаимодействовал в самом начале. И он предложил прилететь на хайринг эвент в Варшаву и там провести мой онсайт.

Варшава, Польша:

Амазон оплатил билеты и гостиницу, а также обещал компенсировать все расходы на еду и транспорт. Забегая вперед скажу, что я не стал заниматься возмещением. Амазон снял хороший отель Marriot, в 5 минутах пешком от другого отеля, в помещениях которого они проводили онсайт. В день интервью пришел на место проведения, немного пообщался с HR и для меня на все время была выделена одна переговорка.

Интервью:

1) Кодинг.

Сначала Leadership Principals на 15-20 минут. Затем задача на кодинг у доски. Не вдаваясь в детали, я себе ставлю за эту секцию незачет, так как фактически задачу не решил, долго выяснял все пограничные кейсы, сделал функции хелперы и что-то успел написать из решения. Ну и стоит добавить, что я фактически не готовился к кодингу, так как шел на systems и делал упор на linux и system design. Такое начало меня немного демотивировало вначале, но сумел собраться.

2) Менеджерская секция

Пришел хайринг менеджер той команды в которую я подавался и целый час гонял меня по Leadership Principles, я реально утомился придумывать истории на ходу или вспоминать заготовки. Но если бы я не сделал упор на LC накануне, я бы точно на ходу придумал абсолютно все истории. Почти все вопросы начинались с фразы "Tell me about a time when you ...". Они ожидают конкретную историю, рассказанную с применением STAR. Никаких гипотетических рассуждений, только история из прошлого. На некоторые вопросы, я отвечал, что не сталкивался с такой ситуацией и у меня нет истории. Тогда он доставал из копилки другой вопрос.

3) Systems/Troubleshooting

15-20 минут вопросы на Leadership Principles.

Очень абстрактная беседа, больше на здравый смысл и некоторую информированность о Linux. Просто рассказал, на примере Яндекса, как я работал SRE. В целом тоже самое описано в SRE Book от Google, что-то рассказывал оттуда. Прошла легче всего и очень непринужденно. Эту секции проводил инженер, который был на втором телефонном интервью, очень приятный парень. Я бы даже сказал, что это было больше похоже на простую беседу на кофепоинте.

4) Систем дизайн.

15-30 минут вопросы на Leadership Principles.

Попросили спроектировать систему бронирования отелей. Собрал требования, выделил 3 основных функции сервиса. Секция была супер абстрактная, никаких уточняющих вопросов, никаких деталей. Я много говорил, он без остановки записывал в компьютер.

Например:

Я: мы могли сами готовить базу, заниматься шардированием и репликацией, или взять готовую DynamoDB в AWS.

И: отличное решение c DynamoDB, давайте использовать ее.

Тут я сразу понял, что от меня хотят. В итоге я собрал сервис из других сервисов AWS. Информацию о бронировании отелей по его же условию надо было брать во внешнем API. Мы не погружались в детали ни в одну из частей схемы.

Потом за 10 минут до конца, он предложил мне задать вопросы о работе в Amazon. Хотя по дизайну там было еще куча чего обсудить. Время было уже в районе 17 часов, я предполагаю, что он уже просто устал за целый день и формально подошел к вопросу.

Резюме.

50% всего интервью занимают Leadership Principals, вопросы выложенные здесь https://leetcode.com/discuss/interview-question/437082/Amazon-Behavioral-questions-or-Leadership-Principles-or-LP подходят на 100%, прям слово в слово. Можно смело готовиться по ним. Плохо пройденная секция по кодингу оставила осадочек, так бы я сказал, что все было неплохо.

Через неделю после онсайта, мне пришло письмо от рекрутера, что мол поздравляют и готовы предложить мне присоединиться к их команде AWS RDS. Еще через пару дней был звонок с рекрутером на тему релокации и бенефитов. Про зарплатные ожидания и про цифры не говорили.

3 или 4 января мне пришел официальный оффер с цифрами, я ответил, что ожидаю решения хайринг комитета гугла и только после этого приму решение.

О переговорах с Амазоном и Гуглом, расскажу в третьей части, после рассказала об онсайте с Гуглом.

**@EvgenyPanov**

**Предыстория:**

Подавался через рефер от своего друга в Google, на позицию SRE. Сначала был init call с HR, которая спросила про опыт работы и пробежалась по тесту, где даются варианты ответов. Тест по Google/Python ответил на 100% правильно, по Linux Internals наверно процентов на 50%. Меня спросили по какому направлению я хочу собеседоваться SRE-SWE или SRE-SE. Я выбрал SRE-SE. После этого мне назначили телефонное интервью.

Начали с вопросов по Linux, дальше был кодинг и 10-15 минут ответы на мои вопросы. Кодинг приближен к рабочим задачам, надо было побегать по /proc/ собрать некоторую информацию и вывести результат. Google это называется practical coding, нужно написать аккуратный, читаемые код без багов.

После телефона прошла примерно неделя, мне позвонил HR и дала фидбек, что кодинг был отлично, а вот linux internal надо улучшать. Она поинтересовалась, как у меня продвигаются дела с Amazon, я сказал, что жду онсайта. После чего я примерно еще неделю ждал, когда назначат второй телефон, но это не происходило. Я пинганул HR и она снова связалась со мной по телефону и сказал, что хайринг менеджер принял решение не проводить второй телефон и сразу звать на онсайт. Далее меня передали другому HR и мы запланировали онсайт сразу через 2 дня после онсайта с Амазоном. На выбор мне было предложено 3 страны: Ирландия, Швейцария, Германия. Я выбрал Швейцарию.

В итоге сразу после онсайта с Амазоном, я перелетел из Варшавы в Цюрих.

**Leadership.**

25-30 минут leadership questions. Задают 2 типа вопросов:

- Привести пример из прошлого, как я действовал в той или иной ситуации

- И как бы я действовал, если ситуация случится в будущем.

После Амазона, показалось легко, на некоторые вопросы давал заготовки для Амазона.

**Troubleshooting.**

Есть некий сайт, твой коллега говорит тебе, что иногда сайт грузится дольше, чем обычно, нужно найти причину.

Сначала задавал вопросы на счет исправности клиентского оборудования, описал как бы убедился в том, что с компом и сетью и на клиентской стороне все ОК. Далее решили, что все ОК и надо копать инфраструктуру.

Интервьювер нарисовал на доске график (время выполнения запросов). Я задавал вопрос на счет того, какие инструменты у меня есть. Например, могу ли я, где-то посмотреть список инцидентов, которые создала тех. поддержка или еще кто? Может быть проблема уже известна и ей занимаются.

Есть ли у меня дашборд с релизами и могу ли я сматчить выбросы на графиках с временем релиза? И если да, то может быть дело в релизе и его надо откатить.

Есть ли у меня система трейсинга запросов, могу ли я попросить своего коллегу, выполнить запрос добавив какой нибудь query параметр в строку браузера, чтобы получить trace ID и далее отправить его в систему и получить расклад по latency на всех уровнях системы. Интервьювер сказал, что это было бы слишком просто, поэтому продолжаем копать дальше руками.

Поговорили про мониторинг, поддерживает ли мой мониторинг запросы, чтобы я мог отделять локации, проекты, группы, сервера на различные части? Поговорили про сеть, решили, что с сетью проблем нет.

Далее я предложил найти корреляцию между базовыми метриками на серверах и графиком latency. Пришли к выводу, что всплески совпадают с графиками CPU. Далее интервьювер разложил графики группы серверов с application на отдельные и представили, что на машине #3 потребление в разы выше, чем на остальных.

Я спросил, нужно ли мне убрать машину #3 из кластера, чтобы она не влияла на продакшен? Интервьювер сказал нет, так как 2 оставшиеся не справятся с нагрузкой.

Пошли на машину по ssh и начали смотреть, что там. Поговорили о тулзах Линукса, top, free, pidstat, iotop. Я предложил почитать логи приложения, там конечно все ОК. Базовые логи сервера тоже в порядке, с железом проблем нет.

Посмотрели с помощью iperf top, что делает приложение, затем предложил снять статистику и построить flame graph, чтобы найти горячие функции. Интервьювер сказал, что мы выяснили, что проблема в функции, которая работает с базой данных. И в этот момент закончилось время, дальше капнуть не успели.

**Систем дизайн.**

Приишел парень, тимлид, 10 лет в гугле. Попросил спроектировать key/value storage из которого только читают. Постановка задачи звучала ровно так.

Я начал задавать уточняющие вопросы:

1) Какой размер ключей и значений, какой тип данных? key 1КB string и val 10KB binary

2) Сколько всего пар будем хранить в нашем сторадже? 5 млрд.

3) Какая нагрузка на чтение? 100к запросов в секунду.

4) Какое железо у меня есть? 32CPU/256GB RAM/10 Gbit net/ 10 \* 6 Tb HDD

5) Какие к сервису предъявлены SLO? 95% req < 50 ms и availability 99.99%

Далее начали считать цифры, сколько всего нужно места для хранения такого объема, сколько сети. На это было потрачено больше всего времени, было разрешено юзать калькулятор в телефоне. В итоге вышло, что для ключей надо 51.2 Tb и для значений в районе 500Tb. Затем начался разговор про то, как будем хранить на диске или в памяти. Далее речь зашла про время позиционирования головки жесткого диска и количество iops на каждый диск. Я предложил для старта хранить все в RAM. Расчеты показали, что при текущем железе, надо 2.3к серверов, предложенной конфигурации.

Далее сказал, что мы можем сэкономить потребление памяти и часть данных хранить на диске или использовать SSD, вместо HDD. Сказал про LRU кэш и на этом время закончилось.

**Обед.**

Пришел другой сотрудник с которым пошли пообедать в их столовку, пообщались про работу в Google, про жизнь в Цюрихе.

**Linux Internals.**

Пришел инженер 8 лет опыта в Google. Начал с простого вопроса про различие симлинков и хардлинков. Потом сказал, что есть веб сервер, который отдает файлики, нужно подменить содержимое файла и минимизировать простой. Я предложил использовать SIGHUP для reload конфигурации или использовать симлинки. Далее он копал в глубь и просил придумать еще способ. Я предположил, что можно подцепить к процессу с помощью GDB, открыть новый файловый дескриптор и с помощью системного вызова dup, заменить старый дескриптор на новый.

Затем поговорили про жизнь процесса, fork, exec, zombie process. Потом снова вернулись к веб серверу, как бы я в проде решал эту проблему, если бы писал сервер сам с нуля. Я предложил два варианта, использовать менеджмент порт для взаимодействия с веб сервером по gRPC и отправлять ему команды. И второй варианты был использовать etcd или подобную систему, чтобы мое приложение само ходило за актуальной конфигурацией. Было, что-то еще, но уже позабыл.

Большую часть в этой секции я позабыл и не смог восстановить в памяти, были еще вопросы про kernel/user space, сигналы и что-то еще.

**Кодинг.**

Пришел мужик предпенсионного возраста, долго не мог настроить ноут для собеседования, очень тихо говорил, было сложно разобрать. Дал задачку написать на python аналог программы pstree. Долго проговаривали решение, своими подсказками он скорее меня сбивал с толку. Относительно остальных интервьюверов он выглядел супер уныло и не заинтересовано. В итоге начал писать код, на середине программы кончилось время. По-сути задачу не решил.

Потом выяснил у него, что он работает чуть больше года, и вероятно у него мало опыта собеседований. Последняя секция с ним меня сильно расстроила, потому что до этого все было отлично.

**ОФФЕР**:

Через пару дней мне написал мой HR, предложил созвониться. Меня это дико напрягает, когда они не пишут в письме тему разговора и какой то краткий итог. Потому, что до момента звонка ты не знаешь да/нет, это очень нервирует. Амазон сразу писал в письме, а по телефону давал детали.

Попросил мой фидбек по секциям, затем дал фидбек со стороны Google. Сказал, что хайринг менеджер передает мое дело в хайринг комитет. Так как это было преддверие нового года, то рассмотрение было назначено на 6 января.

7 января мне позвонили и сказали, что не успели рассмотреть и перенесли на 8 января и вечером 8 января мне позвонили и сказали, что хайринг комитет одобрил мою кандидатуру и теперь надо дождаться официального оффера. Я спросил у своего HR, а как же кодинг? По моему мнению я считал, что завалил его. На что он мне сказал, что комитет посмотрел фидбек с телефонного интервью и там был хороший отзыв и им этого хватило для принятия решения.

В итоге, 10 января вечером, я получил официальный оффер и далее начались переговоры c Амазоном и Гуглом.

UPD: Google перед выставлением оффера сразу спросили цифры от Амазона и прислали сразу улучшенный оффер.

Переговоры после двух офферов от Амазона и Гугла.

После того, как Гугл выкатил свой оффер, я написал хайринг менеджеру из Amazon в LinkedIn, что мол так и так, Гугл дал такой оффер. Он ответил, что дайте нам время, мы сейчас улучшим наш. В итоге через 2 дня Амазон присылает улучшенный на 25% оффер.

После чего, я написал в Google, что мне нужно еще пару дней, так как Амазон выставляет новый оффер. На что получил ответ, что потом покажите его нам, мы попробуем улучшить свой еще раз.

В итоге Google еще раз апнул свой и я его принял.

**РЕЗЮМЕ:**

Теперь мое мнение такое, что если собеседоваться в FAANG, надо идти сразу 3-4 компании.

Чтобы онсайты были максимально рядом и не пришлось затягивать принятие офферов. Судя по Google и Amazon, на них сильно влияет, что кандидат параллельно успешно проходит интервью в другие компании и другой оффер помогает получить лучший результат.

**@matveevtut**

На вопрос "как подготовиться к алгоритмам с нуля?" всегда отвечаю, что нужно первым делом пройти оба курса Седжвика, прорешав все задачи. Без задач толку будет сильно-сильно меньше. Если всё по графику проходить, т.е. каждую неделю с пн по ср разбирать лекции, потом переходить к тесту и задаче, то можно уложиться в то время, которое на курсере написано - 6 недель на первый курс и столько же на второй. В таком графике вполне реально совмещать с фуллтайм работой. Когда я проходил, у меня задача к неделе занимала обычно один целый выходной с утра до вечера.

Седжвик поможет раз и навсегда разобраться во всех нужных структурах данных и алгоритмах, их сложности и научит комбинировать несколько структур/алгоритмов для решения конкретной задачи (именно для этого и нужны домашние задания). Главный недостаток - там плохо раскрываются парадигмы типа Divide & Conquer, Dynamic Programming, Greedy Algorithms. Да, после курса можно легко применять изученные алгоритмы, но взять и придумать новый алгоритм на DP будет не понятно как. Тут приходит на помощь специализация Тима Рафгардена (https://www.coursera.org/specializations/algorithms), там материала ещё на 4 месяца. Я сейчас на последнем курсе из специализации, и пока не было ни одной hard задачи на leetcode, которую я не мог бы решить. Так же курсы Рафгардена учат мыслить абстрактно, доказывать корректность работы твоего алгоритма, писать псевдокод на бумажке. Это всё, имхо, очень ценные навыки для прохождения интервью.

Если у вас нет фуллтайм работы, то можно взять оба курса сразу, но лучше сначала убедиться, что можешь справляться с одним и остаётся свободное время.

Рафгардена - нет, там все домашки сводятся к тому, что ты реализуешь алгоритм как хочешь, прогоняешь его на огромных входных данных и копируешь ответ

А насчёт джавы - я бы посоветовал её не бояться. У Седжвика только минимальное её подмножество, которое понять очень легко. Т.е. не будет всяких ООП-паттернов типа FacadeFactoryAdapterProxy, фреймворков типа Spring. Ваш код будет сводиться к одной функции, решающей задачу. Напротив, курс позволяет в деталях увидеть, как реализованы некоторые части многих современных рантаймов, напрмер, оптимизация строк похожа на то, что есть в питоне.

**@svolodko**

Вставлю свои пять копеек. Я согласен с тем, что оба курса очень хороши, что они не конкуренты, и что лучше начинать с Седжвика. У него действительно сразу после каждой идеи идёт её конкретная имплементация, и таких красивых и лаконичных имплементаций ещё поискать. Вставка в бинарное дерево поиска - это сказка!

Но при этом у Седжвика действительно нет почти ничего про DP, и про Greedy algorithms он только обмалвливается. И тут на подмогу приходит Рафгарден! У него же большой курс поделён на 4 маленьких, так вот, третий из них - как раз про DP & Greedy algorithms design paradigm. И лучшего объяснения этих подходов я не встречал, автор очень логично подводит к ним. И когда у меня появляется задачка, и я схожу не вижу возможности разбить её на подзадачи, то я начинаю мыслить ровно как он учил, слово в слово: What if we have the optimal solution?! How does it look? What if little birdie could tell us the right choice?

Возможно, стоит совмещать Седжвика с задачами на литкод. Берешь лекцию Седжвика про деревья, прослушиваешь, чуть ли не копируешь имплементации, что Седжвик демонстрирует - и вперёд на литкод по тэгу Tree. Начинаешь с лёгких, та же вставка в дерево, поиск в глубину, слияние двух деревьев, и тд. К слову, я таки забил на домашние задачки Седжвика с какого-то момента. Те же проблемы на литкоде более приближены к вопросам с интервью + там меньше воды + не надо ставить Java & IDE :) Хочешь - решай на питоне, хочешь - на C#.

**@abcdmitry**

Только вернулся с онсайта в Амазоне в Мексике. Видимо я один из последних, кто смог прорваться на онсайт.

Подавался на SDE, прошел OA, рекрутер предложил попробоваться на SDM, но после телефонного интервью с хайринг манагером сообщили что SDM предложить не смогут.

Атмосфера очень доброжелательная, создалось впечатление, что они очень заинтересованы в том, чтобы я справился. Задачи уровня easy-medium, никакого харда с литкода не было.

Хотя я и не спрашивал, сказали, что бизнес-трипы в US / Canada идут вдовесок, надо больше общаться с командами лично.

**Из приколов:**

Интервьюеры видимо между собой обменивались впечатлениями, и заметили, что я подхожу к решению задач по схеме из CtCI. Поэтому третий интервьюер меня потроллил - озвучил задачу, без примеров, без ничего, и говорит "That's it, start coding now :)"

У другого интервьюера была задача на тему музыкального ритма, примеры он прохлопал в ладоши и промычал. В общем подошел креативно.

Баррейзера было вычислить несложно. На одном из бехейвиорал вопросов я слишком сильно наимпровизировал, он мне сообщил, что я в одну историю включил слишком много LPs, и попросил помочь ему разбить кейс на части.

Спросил у хайринг манагера в лоб, почему отказали в SDM. Получил такой же прямой ответ, что нет опыта с построением систем масштабов Амазона. Я специально уточнил, повлиял ли мой английский или interpersonal skills? Он ответил что совершенно нет, с этим все в порядке.

Так что видимо скакнуть из восточной европы сразу на SDM абсолютно реально, надо только иметь стратегию, и фокусироваться на тех моментах, которые они хотят услышать.

**Aleksei Eraskin**

Когда задают вопросы не указывают к какому принципу он относится. И думаю на интервью уже не будет времени это быстро сопоставить. Я взял top 20 вопросов от Dan Croitor и список на литкоде (<https://leetcode.com/discuss/interview-question/437082/Amazon-Behavioral-questions-or-Leadership-Principles-or-LP> ) и их отрабатывал. Одна история может покрывать несколько LP. Я старался упоминать что вот такая то ситуация это пример такого то принципа если была возможность. Принципы пришлось наизусть выучить, поразмышлять над ними. Повспоминать свои жизненные ситуации. Думаю это было в целом полезно для меня даже в отрыве от интервью. Все равно на интервью не все пошло так гладко. Пришлось импровизировать местами. Пару раз я с учителями на italki и на pramp делал сессии по этим вопросам.

Т.е. в итоге по behavioral это такой параллельный процесс - ты готовишь/вспоминаешь свои истории по вопросам. Постепенно у тебя их набирается 10-20. Ты можешь про них долго говорить с кем то или сам с собой. Отдельно учишь LP принципы, а потом их сопоставляешь. И когда рассказываешь какую то историю уже можешь их соотнести.

**@cepefagel**

Я сначала попробовал писать рекрутерам в линкедин, все отвечали что сейчас есть вакансии в Таллине и подавайся через сайт. Я зааплаился сразу в 3 локации - Прагу, Таллин и Дублин. В Дублин почти сразу пришёл отказ. В Таллин через пару дней рекрутер вышел на связь и засетапил первый звонок, потом был второй звонок с менеджером проекта на который была вакансия. Сказали что ещё неделю будут общаться с кандидатами и потом решат с кем двигаться дальше, но через 1,5 недели вышел на связь другой рекрутер из Праги и засетапил первое онлайн техническое, а после него уже позвали на онсайт. Однако у меня Майкрософт стек и скорее всего это дало небольшой буст

Продолжение про сам онсайт и онлайн тест в Microsoft Prague, Dynamics 365 marketing team

Онлайн тест проходил как обычно - видео звонок через MS Teams длительностью 1ч с разработчиком из команды в которую я подавался (так вроде только у Майкрософта). Интервьюер попросил рассказать про мой опыт, потом спросил как мне будет удобнее решать задачку через онлайн тулзовину или пошарить экран с моей IDE, я вибрал второй вариант с Visual Studio. Сама задачка была классической - сделать reverse для односвязного линкед листа, фоллов ап вопрос был сделать метод иммутабельным (возвращать новый список, а не делать перестановки в исходном) и как бы я его тестировал.

Онсайт в Праге состоял из 4 раундов без ланча и систем дизайна, каждый длительностью 1 час

1) Интервью с Senior Service Engineer, самое короткое ( 40 минут ) и включало в себя небольшой тур по офису, интро про команду и проект, вопросы про опыт и мою роль на проектах, немного обсудили как было настроенно CI & CD на моем предыдущем проекте (у них использовались такие же технологии), было пару вопросов про Azure. Кодинг задания не было

2) Кодинг интервью с Senior Software Engineer, интервьюер прошелся по резюме и поспрашивал про технологии с которыми я работал, спросил про pros & cons SQL vs NoSQL БД. Кодинг задание уровня медиум литкода - написать 2 метода: один для сериализации дерева ( не бинарного, а дочерних узлов может быть много ) в строку и второй для десериализации строки обратно в дерево. Я немного поплыл с десериализацией, но после небольшой подсказки вроде справился.

3) Кодинг интервью с Senior Software Engineering Manager, начали со стандартного вопроса про опыт и прошлись по резюме. Задание - отсортиовать книги по частоте рекомендаций, в ходе выяснения требований оказалось что возвращать нужно не все, а только топ К шт - вообщем не просто сортировка хеш мапы, а оптимальнее было заюзать Priority Queue

4) Coding/Behavioral с Principal Group Software Engineering Manager, опять вопрос про опыт и пару bh вопросов по типу чем ты можешь быть полезен Майкрософту и кем ты себя видишь через 5 лет) само задание - написать максимально оптимальный метод для проверки является ли число простым, но без заранее вычисленного массива простых чисел

в результате получил оффер на позицию

Title: Software Engineer II

Level: 61

Основное это английский язык, легко будет переехать в США через год, т.к. часть команды сидит в Сиэтле ну и если учитывать уровень/стоимость жизни то компенсация была немного выгоднее.

Подготовка: где-то чуть меньше 5 месяцев фулл тайм, но я не особо напрягался

2 курса Седжвика на курсере

немного ctci

курс grokking the system design interview

240 литкод задачек ( 40% easy, 50% medium, 10% hard )

Всем привет👋

Хочу продолжить инициативу написания о собственном опыте прохождения в компании вроде FAANG и поделиться мыслями по всему процессу начиная от подготовки и заканчивая первыми неделями работы в компании. Считаю это очень важно, поскольку это мотивирует других людей и если бы в свое время один человек не поделилась своим опытом то скорее всего этого чата не было б!

Когда я начал готовиться?

Я учился в физико-математическом лицее последние два года школы, а перед лицеем и во время обучения в нем я ездил на олимпиадкы по математике и информатике (ничего серьезного, еле занимал третье место на области). Можно сказать, что уже тогда я начал готовиться и мог развернуть строку за О (n) времени и О (1) памяти, знал (но не мог бы реализовать) о всяких листах, стеках, очередях, деревьях и графах. Сейчас я понимаю, что эти знания можно было бы получить за неделю хорошего обучения. Так как на первых трех курсах разговоров за то, чтобы подаваться в FAANG не было, а если и были какие-то истории то это только о супер крутых чуваков, то и мотивации готовиться тоже не было. На четвертом курсе я уже начал решать всякие литкоды, интервюбиты и хакерранки и смотрел разные плейлисты на ютубе по алгоритмам и структурам данных, но все равно это не было систематически. Я мог решать задачи все выходные, а потом забить на несколько недель и не решить ни одной или посмотреть плейлист на ютубе и не возвращаться к задачам несколько месяцев.

В целом по бэкграунду у меня следующие результаты:

0️⃣ курс на [stepik](https://stepik.org/course/217/syllabus)

1️⃣ просмотрено большое количество видео на ютубе из которых я могу выделить два плейлисты [туть](https://www.lektorium.tv/course/22823) и [туть](https://www.lektorium.tv/course/22843)

2️⃣ много статей в каждой из которых рассказывают как решать какую то задачу

3️⃣ leetcode (212 solved: 72.6% easy, 25%, 2.4% hard😅)

4️⃣ прочитал [Grokking the System design interview](https://www.educative.io/courses/grokking-the-system-design-interview?aid=5082902844932096&utm_source=google&utm_medium=cpc&utm_campaign=grokking-manual&gclid=EAIaIQobChMI8s23oNTO6AIViLbtCh3MXQaMEAAYASAAEgIHv_D_BwE) Там как по мне самое важное - прочитать словарь терминов, чтобы знать какие есть решения, и какую-то одну статью о построении дизайна, чтобы запомнить паттерн, а дальше все идет по накатанной. Но пройти весь курс также довольно просто.

5️⃣ Прочитал первые 150 страниц [cracking the coding interview](https://www.amazon.com/Cracking-Coding-Interview-Programming-Questions/dp/0984782850) , оттуда я понял по какому принципу нанимают сотрудников в FAANG

Эмпирическим путем я пришел к тому, что ничто меня так не мотивирует готовиться к интервью как уже запланированное интервью. Из этого не тяжело сделать вывод: больше интервью ~ лучшая подготовка. И здесь понеслась:

❌ Bolt

Подавался в апреле 2019 без рефера. Как такого разговора с HR не было, мне прислали онлайн задания в котором надо было решить одну задачу за определенное время. В задаче надо было развернуть все слова в тексте. После этого мне назначили онлайн интервью. На нем сказали написать функцию в которую передается строка и функция должна возвращать true если в строке все уникальные символы встречаются с одинаковой частотой. Я написал код который работал и выполнял то, что нужно было, а после этого интервьюер сказал модифицировать решения так, что теперь функция должна возвращать true если символы встречаются с одинаковой частотой или только один символ может иметь частоту на 1 больше.

Результат: сказали лучше подготовиться и податься еще раз через некоторое время.

✅ Grab

Подавался в июне 2019 с рефером. Сначала был онлайн есесмент на котором было две задачи: [раз](https://pasteboard.co/J2fs9Al.jpg) и [два](https://pasteboard.co/J2frlja.jpg). А потом три технических интервью.

На первом сказали написать функцию которая получает строку на вход и должна вывести эту строку в консоль следующим образом: одно слово вертикально, одно горизонтально и так пока слова не закончатся. Плюс были behavioral questions. На втором интервью дали две задачи [Kth smallest element](https://leetcode.com/problems/kth-largest-element-in-an-array/) i [missing element](https://leetcode.com/problems/missing-number/) . Для каждой сначала надо было рассказать все варианты решения задач, а затем выбрать произвольный для реализации. И на третьем интервью был устный траблшутинг: упал какой-то сервис и я рассказывал что бы я делал, а на мои действия интервьюер отвечал это как-то помогло узнать больше о проблеме или нет. Потом дал еще задачу в которой был

список http запросов в формате: ip машины которая осуществляет запрос и ip машины на которую осуществляется запроси по этому нужно было построить граф всех вызовов. А затем в конце еще немного разговор о жизни.

Результат: дали оффер я отказался.

✅ Microsoft Belgrade

Подавался в августе 2019 без рефера. Сначала был разговор с HR. Затем онлайн ессесмент с простыми задачами. После чего позвали на онсайт в Белград. Там было четыре интервью подряд. На первом попросили рассказать о своем проекте и его архитектуру и дали задачу на проверку являются ли две строки перестановкой некоторого числа символов и задачу на нахождение всех подстрок строки А которые есть перестановкой символов из строки В. На втором интервью несколько вопросов по резюме и задача на восстановление бинарного дерева с матрицы предков. Задачи с третьего интервью я не запомнил. А на четвертом интервью были вопросы о жизни, обо мне и насколько я отвечаю культуре компании.

Результат: готовы были дать оффер, но я не захотел.

✅ Microsoft Prague

Подавался в августе 2019 с рефером. Здесь следует пояснить: в отличие от большинства компаний которые нанимают в компанию, Microsoft нанимает людей в конкретный офис, поэтому у меня была возможность проходить процедуру интервью практически параллельно в два разных офисы Microsoft. Причем если вас будут готовы принимать в два офиса то вам сначала придется выбрать офис, а потом вам дадут для рассмотрения сам оффер.

Сама процедура прохождения была следующей: разговор с HR, онлайн техническое интервью на котором дали одну решить задачу. Задача заключалась в том, что на вход дается список строк представляющих путь к директориям, надо было написать функцию которая создаст на компьютере все директории указаны в списке и соответственно директории которые есть на пути к ним. После этого пригласили на хайринг ивент в Прагу. Там было четыре интервью. На первом нужно было написать деадлок, говорили немного о многопоточность, затем перешли к дизайну классов для мессенджера в котором могут быть различные типы сообщений: текстовое, аудио и т.д. Второе интервью было на системный дизайн: дали задизайнить мессенджер для общения между двумя людьми. На третьем интервью были вопросы о жизни, мою мотивацию, опыт и тому подобное. А на четвертом была алгоритмическая задача.

Результат: дали оффер и я принял.

❌ Google

Подавался в сентябре 2019 без рефера. Написал на емейл очередного HR без надежды, что мне кто-то ответит, но все сложилось гораздо лучше чем я ожидал. Мне ответила рекрутерка и мы договорились об онлайн звонке. На нем она сначала рассказывала стандартные вещи, а потом задавала технические вопросы с вариантом выбора одного и несколько ответов. Не на все я отвечал правильно, но это не помешало получить техническое онлайн интервью. На нем дали задачу на динамическое программирование, в ней надо было сказать доскакает ли дракончик до финиша имея возможность скакать на любое расстояние от 1 до К и на пути есть кактусы на которые становиться нельзя, весь путь представлен булевым массивом, где і-тое значение true если есть кактус. После этого пригласили на онсайт в Мюнхен. Там было 5 интервью + обед. Список всех задач можно найти [здесь](http://bit.ly/g-i-p).

Результат: не дали оффер (закончились наверное😄)

✅ Amazon

Подавался я в ноябре 2019. Меня посоветовал рекрутеру в LinkedIn знакомый парень. После этого рекрутер написал мне и сбросил онлайн задания в котором надо было решить задачу [раз](https://pasteboard.co/J2fYJPI.jpg)и задачу [два](https://pasteboard.co/J2fYWQk.jpg) . Решать их приходилось на платформе от Amazon. Причем в конце давалось 15 минут для того что описать алгоритм и сложность решения. После этого меня пригласили на хайринг ивент который проходил в Киеве, поэтому ехать никуда не надо было. На самом ивенте было 4 интервью. Каждое интервью проходило следующим образом: 3-5мин на короткий интродакшин по адженде интервью и о себе, дальше 18-22мин. на 2-3 вопроса по их лидершип принципам, 20-25мин. на алгоритмическую задачу (одну) и в конце 3-5 мин на вопросы. Одно исключение из этих правил так это то, что на одном интервью вместо алгоритмической задачи было задания на системный дизайн - задизайниты gmail. Была задача на поиск количества вхождения некоторого target элемента в отсортированном массиве. Похожа задача [здесь](https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/) .

В следующей задачи дается список объектов, которые представляют модули для установки и надо было написать функцию которая их устанавливает, но при этом у модулей есть модули от которых они зависят и их надо установить первыми, плюс надо хендлить циклы. Вопросы на лидершип принципы были такие: привести пример когда я работал над таской, что мала вероятность выйти за дедлайны, делал ли я что-то, что выходило за пределы моей ответственности, привести пример когда я вносил изменения в дизайн архитектуры проекта, привести пример когда мне давали фидбек и как я его принимал во внимание. Один раз меня попросили привести пример ситуации которую я не мог вспомнить чтоб случилась со мной, я так и сказал, что не припоминаю, после чего меня попросили привести пример другой ситуации.

Результат: дали оффер, но я не принял.

А сейчас я побуду капитаном и поделюсь некоторыми мыслями:

📌 Всегда следует обсуждать зарплату, даже если у вас нет контроффера. Стоит понять, что зарплата - это далеко не основные расходы для компании на вас как сотрудника; в худшем случае вы ничего не теряете; стоит придумать что-то лучше чем просто написать: "дайте больше денег". Вот три статьи которые помогли мне понять это и много еще: [раз](https://haseebq.com/my-ten-rules-for-negotiating-a-job-offer) , [два](https://haseebq.com/how-not-to-bomb-your-offer-negotiation/) и [три](https://medium.com/@bayareabelletrist/how-i-negotiated-a-software-engineer-offer-in-silicon-valley-f11590f5c656) .

📌 На каждом интервью скорее всего вас попросят рассказать о себе, это одна из немногих вещей к которым можно подготовиться заранее на все 100%, а не придумывать ответ на ходу.

📌 Больше всех кто мне мешал подготавливаться это 🥁🥁 я сам. Если в начале я разделял задачи на легкие и тяжелые, то со временем я начал их делить на те, что я умею решать и не умею, это помогло проще относиться к темам которые раньше казались тяжелыми для изучения.

📌 Часто бывают долгие споры о том какой курс пройти или на какой платформе задачи решать, как по мне то результата от этого не много, поскольку чужой опыт не обязательно будет хороший для вас. Лучше начать проходить любой курс или решать задачи на любой платформе и понять он хороший или нет.

📌 Количество решенных кем-то задач на leetcode также особо ни о чем не говорит, более того, порой эта информация может быть даже вредной, поскольку узнав, что кто-то решил на leetcode условно 100 задач перед тем как начать проходить собеседования, может быть для вас барьером для того чтобы начать подаваться в FAANG раньше.

📌 FAANG компании если сомневаются в человеке то пользуются такой логике: для них лучше не взять хорошего сотрудника, чем взять плохого, ибо первое стоит намного дешевле чем второе. поэтому когда я проваливал интервью, то всегда думал, что я как раз тот квалифицированный которого не взяли из-за сомнений, а это как результат не давало унывать.

📌 Хорошо подумайте хотите ли вы в ФААНГ, ведь вместе с плюсами есть и много минусов. И если вы начнете готовится у вас не должно закрадываться сомнения стоит ли тратить на это время или нет, на этот вопрос вы уже должны себе ответить.

📌 Не все интервьюеры которые проводят интервью сами бы прошли все интервью заново.

📌 Надо научиться понимать интервьюера и показывать себя с той стороны которой он хочет увидеть. Я не имею в виду обманывать, здесь скорее делать акцент на правильных вещах, поскольку время ограничено, а интервьюер за это время должен убедиться что вы тот, кого стоит нанять. Поэтому перед тем как что-то сказать на интервью я часто задаю себе вопрос: "Поможет ли мне эта фраза получить оффер".

Я уже четвертую неделю работаю в Microsoft в Праге. Переехал я из Киева и на данный момент не жалею о смене работы. В общем весь процесс от подписания оффера к перелету занял 6 месяцев. При этом все это время я работал на работе и все, что от меня требовалось так это регистрация на различных сайтах компаний которые помогали и помогают еще сейчас с релокейтом.

Если у кого то остались вопросы, то найти меня можно практически везде под ником mkotsabiuk. Если хотите встретиться в Праге после карантина, то я буду только рад.

Peace✌️

@Pirat1992

### Предыстория

Я получил диплом бакалавра в области программной инженерии в харьковском политехе. Один раз был на личной олимпиаде по спортивному программированию и один раз на командной, но, будучи совершенно неготовым, пролетел как фанера над Парижем. Мысли о работе в лидерах мирового рынка у меня конечно возникали, но как что-то отдаленное и во время учебы я даже не пытался попасть на стажировку туда, ограничившись устройством в местный GlobalLogic после третьего курса.

### Неудачи

**Бестолковые**

Часть моей карьеры я был автоматизатором. И именно в такой момент я решил попробовать. Промониторив вакансии на сайте гугла я нашел вакансию плюс-минус под мой профиль и отправил своё резюме. На удивление, мое резюме чем-то привлекло рекрутера и он вышел на меня. Надо сказать, что я тогда придерживался той точки зрения, что готовиться к собеседованиям не надо. Если ты не можешь его пройти без излишней подготовки, то и работать в этой компании не получится. О том, что на собеседованиях в FAANGи спрашивают алгоритмические задачи, которые нужно отдельно практиковать, я не знал. Ожидаемо это был оглушительный провал. Я получил легкую задачку на деревья и решил ее с огромным скрипом и кучей подсказок. Ретроспективно я понимаю, что это была разминочная задача, на которую отводилось первые 10-15 минут.

После чего я зарегистрировался на сайте codeforces.com, где полагается решать подобные задачки в соревновательном темпе. Но практически не уделял ему внимания на тот момент.

Следующая попытку я осуществил через пару лет уже в Microsoft. Рекрутер сам вышел на меня и я согласился. Понятия не имею почему я опять не готовился. Возможно полагал, что меня спасет сертификация по C#. Честно говоря, о той попытке у меня крайне смутные воспоминания. Но опять все закончилось на телефонном интервью, причем крайне быстро.

**Внезачетные**

Такими я считаю участие в Google Code Jam и Facebook Hacker Cup. Это соревнования по спортивному программированию. На тех, кто попадают в топы, выходят гугловские/фейсбучные рекрутеры. В сумме я предпринимал пять попыток, но все закончились одинаково - проходом квалификационного раунда и вылетом в первом. Причем один раз я не прошел во второй раунд из-за пропавшего интернета. Было обидно, но тот проход был бы больше совпадением.

**Комедия**

Маленький оффтоп - однажды я подался в гугл через реферрала, которого знал не очень хорошо и дергать его больше, чем один раз мне было неловко. Какой же был мой ужас, когда я осознал, что в резюме указал неправильный адрес электронной почты. Мораль - давайте перечитывать резюме кому-то другому. И особо внимательно следите за контактными данными.

**Толковая**

*Disclaimer: я не буду раскрывать детали задач, так как в письме была просьба этого не делать. Я ничего на эту тему не подписывал, но предпочитаю выполнить эту просьбу.*

Летом 2017 года я начал готовиться чуть серьезнее - прошел этот курс по алгоритмизации, который позволил мне решать большее количество задач и заполнить пробелы в базе. Могу советовать его всем новичкам.

В начале 2018 года я услышал о том, что Amazon проводит hiring event. И вот к нему я готовился уже серьезнее - зарегистрировался на leetcode, стал регулярнее принимать участие в соревнованиях на codeforces и решал задачки еще на других сайтах. Но уровень решаемых задач был крайне невысок. На онлайн-отборе получил две задачи, одну решил, вторую нет. Спустя некоторое время меня все же пригласили дальше. Видно сначала пригласили всех, кто решил две задачи, а потом уже "остаток". Готовиться я продолжил теми же методами - codeforces+leetcode. Вдобавок нужно было подготовиться к двум новым аспектам: behavioral и system design. С первым мне здорово помогла жена - подобрав истории из моего опыта, которые соответствуют разным принципам, ценимым амазоном. Ко второму я готовился крайне мало, понимая, что нужно больше вытягивать задачки, ведь их будет больше.

Было 4 интервью - 3 на задачки + одно на system design плюс на каждом из них была behavioral задачи. Задачи были уровня около medium/easy, какие конкретно уже даже не помню, но я жутко волновался и запорол минимум полтора интервью. Получив ожидаемый отказ я не слишком расстроился ибо увидел прогресс. Но интервью я прошел таки плохо, поскольку на следующий ивент меня позвали только через два года.

### Удачи

**Подготовка**

В моей подготовке наступила пауза, связанная с рождением ребенка, но постепенно через год я начал возвращаться к практике. Купил премиум на leetcode. Старался заставлять себя решать задачи по тем темам, которые у меня плохо получались. Довольно быстро я ощутил, что easy задачи для меня практически бесполезны, ведь решение в них зачастую очевидно и перешел на medium + иногда hard. К августу 2019 у меня было около 60-70 задач на leetcode и, полагаю еще столько же на других ресурсах. Кроме того, я прочитал знаменитую Cracking the Coding Interview, но она мне не дала ощутимой пользы. Возможно, если бы я добрался до нее до прохождения курса и неплохой практики, то мое мнение было бы другим. Гораздо больше мне помог этот канал. Здесь ребята хорошо объясняют как надо мыслить, чтобы дойти до решения и разбирают паттерны.

**Microsoft** ✅

Я узнал о том, что в сентябре 2019 в Таллинне будет проходить Microsoft Hiring Event и подался на него. Онлайн-отбор состоял из 3 задачек. 1 easy, 2 medium. Я решил все три и достаточно быстро получил приглашение на собеседование.

Было снова-таки 4 интервью - 3 coding, 1 system design + немного behavioral. Одно из кодинг интервью было немного нетипично, ведь меня собеседовали не по алгоритмическим задачкам, а по многопоточности. Благо, интервьюер требовал не конкретных жестких знаний, а скорее общего понимания как это должно работать. Два других кодинга были medium уровня. В одном мне повезло, ведь такую задачу я разбирал в самолете по дороге в Эстонию:-). Последняя задача была easy/medium, основная сложность была в edge cases. Эту задачу я решил очень быстро. Собеседующий явно этого не ожидал, поскольку он не знал о чем со мной дальше говорить, о чем мне и сказал. В итоге остаток интервью обсуждал со мной детали работы операционной системы и процессора с потоками. К счастью, мне это было знакомо. System design, как по мне прошел плохо. Я сделал часть, но явно плавал в некоторых аспектах хранения и передачи данных. Плюс потом со скрипом рассказывал о банальной схеме криптования с открытым ключом.

Примерно через неделю я получил оффер, но на крайне малую сумму для себя поэтому отказался. Впоследствии майкрософт попробовал улучшить оффер, но все равно было мало плюс сын еще был слишком мал, чтобы идти в садик и "отпустить" жену на работу. Поэтому я его отклонил окончательно.

**Финальная подготовка**

После произошедшего я продолжил активно готовиться, доведя количество задач к сотне (23 easy + 62 medium + 15 hard). Codeforces я практически отбросил, поскольку уже максимально привык к решению задач в напряженной обстановке. Вариативность же задач, которые не вдавались в ненужные дебри, страдала. Плюс leetcode давал возможность классифицировать задачи по темам и компании, имел площадку для обсуждения задач и я чувствовал больше пользы от него.

**Google** ✅

В очередной раз решил помониторить вакансии на сайте и поотправлять резюме на некоторые из них. Неожиданно мне написал рекрутер относительно вакансии связанной с автоматизацией (Engineering Productivity в их терминах). Я не был уверен, что хочу туда возвращаться, рекрутер же меня успокаивал, что это теперь одна ветка с development и разницы нет. В итоге мы решили сначала пройти первое "телефонное" интервью, а дальше уже решать.

На собеседовании я сразу же вытащил "счастливый билет" - hard задача на динамическое программирование. Я смог ее решить полностью и оптимальным способом. Собеседующий увидел, что есть еще время и предложил мне оптимизировать так решение задачи, чтобы его можно было легко параллелить на несколько машин и это я уже не осилил.

Буквально через пару дней мне сообщили, что можно двигаться дальше. Предстояло решить - хочу я дальше идти на позицию автоматизатора или все же девелопера. Девелоперу предстояло 3 coding interview, 1 system design, 1 behavioral. Автоматизатору 1 coding, 1 coding(easy) + testing, 1 automation testing practices, 1 system design, 1 behavioral. Не в последнюю очередь на мое решение повлияло то, что в случае автоматизации мне придется готовиться более разнообразно и вспоминать давно позабытые вещи. Поэтому решил двигаться дальше с development веткой. Onsite должен был состояться в марте в Варшаве, но из-за всем известных событий собеседование было перенесено в онлайн. Согласно рекрутеру, system design интервьюеров меньше, поэтому было решено, что у меня сначала будет 3 coding + 1 behavior. а потом уже, в случае удачи, 1 system design.

На первом интервью мне попалась задача, даже похожую на которую я не нашел на leetcode. Но уровень где-то medium/hard на DSU + bit operation. Довольно быстро я нашел способ как решать ее оптимально, но дальше почему-то из-за мандража я начал как-то сильно нервничать и в итоге плохо написал DSU, озвучивая вслух правильные идеи. Эмоции были смешанные.

Второе интервью было behavioral. Не было ни одного вопроса из топа, который обычно гуглится. Но и не было масштабных вопросов на 10-15 минут. В основном небольшие по конкретным ситуациям.

Третье интервью. Medium задача с продолжением в hard на тему Trie. Hard не успел дописать буквально одну-две строчки, но всю идею успел донести.

Четвертое интервью. Неожиданно повторилась майкрософтовская история и задача была больше на параллелизацию и ее дизайн. Алгоритмической сложности практически не было.

Уже вечером со мной связались и сказали, что будут назначать system design. Я попросил себе две недели паузы на подготовку и прошел этот курс на 80%. Сам курс отличный и помог мне осознать многие вещи. На интервью же была задача, которая снова-таки не упоминается в топах и ее не разбирали. Но я следовал паттерну из пройденного курса и вроде бы неплохо прошел. Повезло в том, что мне не надо было рисовать, а собеседующий сам уже заготовил картинки. Я только писал текст по требованиям. Но это не правило, возможно вам придется рисовать мышкой.

Вечером после интервью рекрутер сказал, что еще первых 4 интервью было достаточно, чтобы подать мою кандидатуру на Hiring Committee - это комитет, который смотрит на рецензии собеседующих, мое резюме и принимает решение брать меня или нет, кроме того рекрутер упомянул, что он нашел для меня уже вариант команды и если мы с менеджером друг другу понравимся, то тот напишет в мою поддержку письмо комитету.

Отлично, с менеджером связались, он рассказал про команду, возможности развития, задачи, я немного рассказал о том, что меня интересует. Впоследствии я узнал, что менеджер написал это письмо в поддержку.

Вердикта комитета я ждал около недели, он оказался положительный, с комментарием, что behavioral прошло отлично, coding очень хорошо, а вот system design, хоть и соответствовал проходному уровню, но мог бы быть и лучше. В принципе это справедливо, так как опыта в реальном дизайне систем у меня довольно мало. Но оказалось, что решения комитета недостаточно и в нынешних условиях еще нужно одобрения от директора, которое я ждал еще неделю. Итог - оффер в Варшаву на уровень L4.

**Amazon** ✅

Спустя два года снова весенний hiring event. Онлайн-отбор немного видоизменился и теперь нужно было еще описать текстом алгоритм решения задачи и поотвечать на behavioral опросник. Задачи были medium уровня, обе связаны с темой islands на leetcode. Я решил обе задачи очень быстро и тем же вечером получил сообщение о том, что я квалифицировался дальше. К сожалению, из-за коронавируса собеседования проходили по видеосвязи и уже после гугловских собеседований. Как всегда у амазона все интервью начинались с behavioral части.

1. был system design сервиса, который очень часто попадается в топах и примерах. И вот тут пришлось рисовать мышкой. Было не слишком удобно, лучше потренироваться в случае удаленных интервью.

2. Алгоритмической сложности не было, нужно было задизайнить функцию по назначению близкую к production code. Тут я не всегда хорошо понимал, что он от меня хочет. Создавалось впечатление, что ему нужно конкретное решение, которое у него в голове, но при этом он был максимально вежливый и сам признавал, когда меня запутывал.

3. Хард на Trie. Здорово похожий на гугловский, поэтому я справился с ним достаточно легко.

4. Последнюю задачу я не смог найти на литкоде, но подозреваю, что easy/medium. Я сначала решил практически оптимальным путем через сортировку, потом озвучил оптимальный - с использованием max-heap, но не успел его закодить, в том числе из-за проблем с амазоновской системой и моим соединением с интернетом, на чем потерял несколько минут.

На следующий день я узнал, что все 4 собеседующих одобрили мою кандидатуру. Тут проще, чем в Google и не надо ждать одобрения от комитета, директора, так что результат тоже оффер.

**Выводы**

Тут постараюсь осветить субъективные вещи, которые были для меня неочевидны перед всем этим процессом, здесь не будет стандартного "готовьтесь усерднее".

Старайтесь как можно скорее отвязаться от решения easy задач. Да, это практика, да, это приятно решить очередную задачу и поднять свою статистику. Но вы просто теряете на этом время. Решать их приемлемо пока вы совсем не разбираетесь в этой теме. Как только начинает появляться понимание - переключайтесь на medium.

Тренируйтесь решать в некомфортных ситуациях - объясняя у доски другому человеку, на соревновании, не используя IDE. Вообще про IDE стоит забыть во всех случаях, кроме контестов. Так как все эти скобочки, запятые и прочее сами себя на собеседовании скорее всего не поставят и отдебажить "ломом" уже не получится.

Этот совет встречается часто, но я все-таки повторюсь. Озвучивайте даже самое тупое решение и думайте вслух. Если вы понимаете, что вам это дается с трудом - практикуйтесь.

Не стоит бояться неудачи, но готовиться надо, ведь в случае полного провала, компания поставит вас в "черный список" на год-другой. Перед своим основным собеседованием сходите на схожие в другие компании, возможно на неинтересную позицию, локацию, неважно. Главное, чтобы процесс собеседования был похож - решение алгоритмических задачек.

Разбирайте задачи с интервью, вспоминайте процесс и где вы застопорились, чего вам не хватило. Возможно вам нужен инсайт, новый паттерн решения, которые нужно поискать в других источниках.

**Pavlo Koshla**

Собеседование было частью хайринг ивента Киевского. Специализация и левел не обсуждались конкретно. Но вроде как набирали опытных кандидатов. Офера пока еще нету.

Мой опыт: Java backend 8 лет. Последние 3 года работаю с распределенными системами в клауде.

Само интервью состояло из 4 раундов. Две алго задачки, сис. дизайн, OOP. В каждом раунде на саму задачку оставалось минут 20. До этого были LP вопросы. Советую очень серьезно отнестись к подготовке к LP.

Актуально для Амазона. Другие компании могут иметь другие процессы.

OOP - вариант Command/Strategy шаблонов. Нужно было реализовать управление роботом с возможностью просмотра истории команд и undo.

Очень длинное описание (строк на 200). Я начал решать как алго задачку (описание позволяло). Интервьювер не особо мешал вначале. Когда код был более менее готов, он попросил добавить гибкости: типа выполнение каждой команды должно возвращать определенный результат. Тип результата может быть разный. Нужна возможность легко добавлять новые команды со своим типом результатов. На этом моменте до меня дошло, что это OOP раунд, но времени уже особо не оставалось, поэтому я просто рассказал словами как бы я реализовал и добавил немного псевдокода/комментов.

Я честно думал, что я провалил этот раунд. Но может такая задумка изначально была.

Sys, design - Amazon photos (аналог Google Photos). 10м пользователей в день. Вопросы стандартные: как масштабировать, АПИ, модель данных, где и как хранить фото, как и что отдавать пользователю. Интервью проводил менеджер, поэтому упор больше был на фичи чем на техническую часть. Не было особо времени копать глубоко (в 20 минут тяжело детально расписать).

Я предложил API Gateway, разделение upload / get на разные сервисы (upload обычно - длительная операция), координацию через Zookeeper, сервис метаданных (с базой метаданных), S3 как хранилка для фото/thumbnails.

Uploader вначале создает запись в сервисе метаданных, формирует путь к фото, аплоадид и генерит thumbnail. Thumbnails кешируются в отдельном сервисе (Feed, не успели до него дойти), в этом же сервисе генерится view для конкретного пользователя исходя из его подписок/предпочтений.

Getter возвращает либо конкретное фото по ID либо список thumbnails. Был вопрос "как возвращает", ответ: генерит presignedUrl на S3.

Feed хранит thumbnails в отсортированном по времени порядке в памяти / снепшотит во внутреннюю базу.

Дальше особо не успели копнуть.

Задачка на поиск топ К анаграмм в файле / стриме. Нужно было очень детально расписать time/space complexity выбранного решения и других возможных вариантов решения.

Задачка на максимальное число запущенных виртуальных машин в определенный отрезок времени. На вход подавался лог файл с сообщениями о start/stop виртуалки, ID & timestamp. Были вопросы по внутренностям Java и просьба описать возможные проблемы реализации. Это был bar raiser (сам признался).

P.S. У других ребят из ивента опыт был другой и задачки, по моему, ни разу не повторились. Поэтому я не думаю, что подготовка этих конкретных задач сможет дать какие-то плюсы. Как по мне, то во время интервью смотрели больше на то, как и о чем я думал и какие вопросы задавал.

Подготовка.

В 2018 случайно увидел [статью Сергея](https://dou.ua/lenta/articles/google-interview/). Вдохновился и начал задумываться о попадании куда-то туда. Так совпало, что Амазон проводил хайринг ивент +- в тоже время. Я на него подался, но отсеялся на онлайн раунде. С того времени начал более мене готовиться по алгоритмам. Я не могу сказать, что я каждый день решал задачки. По личным причинам приходилось делать паузы в несколько месяцев. Всего подготовка заняла около года в период с октября 2018 по апрель 2020. В 2019 был оферы от booking.com (Амстердам, релокация, senior dev), EPAM (remote, Lead dev), Lohica (team lead), DIGI17 (Vancouver, релокация, senior dev), может еще что-то. Несколько провалов, в том числе онсайт Microsoft Tallinn (провалился на обоих алго задачках, сис. диз. и code review прошли более-менее). Но я всегда старался проходить интервью при любой возможности, даже если не был особо заинтересован в смене позиции/компании. Как по мне, то интервью это отдельный скилл, который требует практики.

Порядок:

львиная часть времени ушла на алгоритмы, +-9 месяцев

+- 2 месяца на сис. диз, но я также читал Designing Data Intensive Applications Reliable Maintainable отдельно, + у меня роли были более менее лидовские на основной работе, поэтому опыт дизайна нарабатывался и отдельно от подготовки.

+-месяц на LP, вспоминал реальные истории (не обязательно из IT) пытался натянуть их на leadership principles

Английский к началу подготовки уже был Advanced, в этом плане было легко.

Ресурсы:

<https://www.pramp.com/> (возможность потренировать интервью бесплатно)

<https://leetcode.com> (мой профайл: <https://leetcode.com/kshp/> )

<https://leetcode.com/discuss/general-discussion/491522/dynamic-programming-questions-thread>

<https://leetcode.com/discuss/general-discussion/494279/data-structure-and-algorithm-study-guide>

Курсы

<https://www.educative.io/courses/grokking-the-system-design-interview>

<https://www.educative.io/courses/grokking-the-coding-interview>

<https://www.educative.io/courses/data-structures-in-java-an-interview-refresher>

<https://www.educative.io/courses/grokking-the-object-oriented-design-interview>

<https://www.educative.io/courses/grokking-dynamic-programming-patterns-for-coding-interviews>

Ютубчик

<https://www.youtube.com/channel/UC9vLsnF6QPYuH51njmIooCQ> (сис.диз, очень рекомендую)

<https://www.youtube.com/playlist?list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O> (алго, очень рекомендую)

<https://www.youtube.com/user/purpongie> (алго)

<https://www.youtube.com/channel/UCmJz2DV1a3yfgrR7GqRtUUA>(алго)

Книги

<https://www.amazon.com/Designing-Data-Intensive-Applications-Reliable-Maintainable/dp/1449373321> (настоятельно рекомендую)