

# Техническое интервью без балансировки скобок

Денис Аникин

<https://xfenix.ru>



Проходит в рамках



# Структура доклада

1 Текущая ситуация

2 Ретроспектива

3 Алгоритмические  
собеседования

4 System design

5 Критика алгоритмических  
собеседований

6 Критика system design

7 Общие проблемы

8 Что делать?

# Денис Аникин

Что я делаю

- работаю в Райффайзен банке
- teamlead в 3 командах
- community lead в Python Community
- fullstack: разрабатываю back на python и front на typescript, занимаюсь devops



<https://xfenix.ru>

# Отдельно про мой опыт интервью

Мой опыт не ограничен техническими интервью, но здесь только про них

- Первый раз участвовал в интервью в 2008 году
- Интервьюировал php-, frontend-, python-разработчиков, devops-инженеров
- Сейчас лидирую группу интервьюеров в банке (6 человек)
- Провел около 150 технических собеседований за последние несколько лет
- Подготавливаю интервьюеров, занимаюсь онбордингом и farewell-активностями
- Как разработчик был на сотнях интервью и многие успешно провалил 😊😊😊

# В общем, как с сырниками не должно быть

Или как с помидорами



**Одна из моих целей —  
показать многочисленные  
противоречия в текущей  
практике**

# Почему я решил вскрыть эту тему?

- Наболело
- Многие разработчики действительно не любят современные собеседования
- Индустрия, на мой взгляд, страдает от этого подхода: мы режем разнобразие в компаниях, оставляем друг другу психологические травмы
- Консенсус индустрии: люби технические собеседования и алгоритмы или «умри»

Опишу текущую ситуацию  
так, как вижу её я

Я решил 1000 задач за 3  
года, потратил 7 лет на  
собесы и наконец попал в  
«рожки и копытца» и  
пишу на джанге ручки уже  
10 год



У мбня цыкл не рамботает,  
мбне нубжен ретрит на 3  
недели, а потом саббатикал



# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (тм)

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (tm)

— Никакой прозрачности процесса **X**

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (тм)

— Никакой прозрачности процесса **X**

— Разве я много прошу?



Interviewed hundreds of candidates · 9mo · ✨

Related **How do software engineers evaluate interviews?**

At [REDACTED] we have a number of criteria that we grade on a 4-point scale. The exact list of criteria is confidential and varies somewhat from role to role. but to give an example that should surprise no one, one of the criteria is “comprehension and communication”, so candidates that struggle to understand what is asked of them or who don’t explain their thought process will score low and those who are able to quickly “de-risk” the problem by asking the right clarifying questions and who are able to guide the interviewer through the process will score high. The key though is that for this criterion and others, interviewers have to back their evaluation with evidence. We can’t just say, “poor communication”.

**Access this answer and support the author as a Quora+ subscriber**

- ✓ Access all answers reserved by [REDACTED] for Quora+ subscribers
- ✓ Access exclusive answers from thousands more participating creators in Quora+
- ✓ Browse ad-free and support creators

[Start free trial](#)

[Learn more](#)

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (тм)

- Никакой прозрачности процесса **×**
- Похоже на олимпиаду **×**
- Разве я много прошу?

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (tm)

- Никакой прозрачности процесса **×**
- Похоже на олимпиаду **×**

- Разве я много прошу?
- Я не олимпиец, я  
пришел делать работу  
(хаха, класть json'ы)

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (tm)

- Никакой прозрачности процесса **×**
- Похоже на олимпиаду **×**
- Низкая культура взаимодействия **×**
- Разве я много прошу?
- Я не олимпиец, я  
пришел делать работу  
(хаха, класть json'ы)

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (tm)

- Никакой прозрачности процесса **×**
- Похоже на олимпиаду **×**
- Низкая культура взаимодействия **×**
- Разве я много прошу?
- Я не олимпиец, я  
пришел делать работу  
(хаха, класть json'ы)
- Хотелось бы приходить  
не очереднярой, а кем-то,  
в ком видят человека

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (тм)

- Никакой прозрачности процесса ✗
- Похоже на олимпиаду ✗
- Низкая культура взаимодействия ✗
- Субъективно ✗

- Разве я много прошу?
- Я не олимпиец, я  
пришел делать работу  
(хаха, класть json'ы)
- Хотелось бы приходить  
не очереднярой, а кем-то,  
в ком видят человека

# И где тут проблема?

Не умеешь, так научись! У нас тут серьезный бизнес (tm)

- Никакой прозрачности процесса ✗
- Похоже на олимпиаду ✗
- Низкая культура взаимодействия ✗
- Субъективно ✗

- Разве я много прошу?
- Я не олимпиец, я  
пришел делать работу  
(хаха, класть json'ы)
- Хотелось бы приходить  
не очереднярой, а кем-то,  
в ком видят человека
- Про субъективность мы  
еще поговорим!



# Как развивалась культура технических интервью

# До 1998 года

Записи почти утеряны, но...

- Просили кодить много сотен строчек кода
- Давали кошмарные тестовые (на недели)
- Вопросы из серии «если вы упали в блендер», «три лампочки, один заход», «почему люки круглые», «утки в пруду»
- А иногда просто общались

х

# После 1998 года

# В дверь заходит FAANG\*-стиль

- Алгоритмические задачи! Наши любимые! От easy- до hard-уровня 😊
- От 1 до 5-6 и более секций
- Божественный system design
- Culture fit, behavioral interview. Если не повезет, то и leadership principles, STAR-методология и т.д.
- Особый свод правил поведения и особенный язык (арго)

\* вообще, уже MAANG или ещё что-то там, кому какое дело, это читать вообще никто не будет

Крупная  
социальная сеть



# Как так вышло?

Это мои рассуждения

— Культура закрытого клуба: соревнование за место среди «лучших»



# Как так вышло?

Это мои рассуждения

- Культура закрытого клуба: соревнование за место среди «лучших»
- Большой успех важной компании

# Как так вышло?

Это мои рассуждения

- Культура закрытого клуба: соревнование за место среди «лучших»
- Большой успех важной компании
- Все стали копировать: мы тоже хотим такого успеха!

# Как так вышло?

Это мои рассуждения

- Культура закрытого клуба: соревнование за место среди «лучших»
- Большой успех важной компании
- Все стали копировать: мы тоже хотим такого успеха!
- Дошло до абсурда: маленькие компании и компании среднего размера стали строить интервью по таким же принципам

```
>>> some_fancy_array = [1, 1, 2, 2, 3, 3, 4, 5, 5]
>>> future_result = 0
>>> for one_element in some_fancy_array:
...     future_result ^= one_element
>>> print(future_result)
4
```

# Обсудим алгоритмические собеседования



**Ты выглядишь как соискатель**

# Из чего состоит алгоритмический блок?

Он же лайвкодинг или вайтбординг

- От одной до бесконечности секций
- Задачи «сделай за  $O(N)$ », которым часто требуется оригинальный способ решения
- Или задачи ближе к бытовому уровню, если повезет (скорее нет, ведь «бытовой уровень» может сильно варьироваться)
- Группы алгоритмов: рекурсивные, разделяй и властвуй, динамическое программирование, жадные алгоритмы, прямой перебор (его не любят как раз, он ведет к  $N^2$  и далее), поиск с возвратом, вероятностные алгоритмы

Я на месте) в кофейне 19:43 ✓



## Что думают интервьюеры об алгоритмах?

Отличный объективный способ, чтобы проверить, как человек:

- Решает задачи
- Задает вопросы
- Разбирается в проблемах

Кроме того:

- Можно посмотреть, как человек себя ведет в стрессовой ситуации
- А ещё, разработчики в основном одобряют эту практику



# Обсудим system design собеседования

RE

Помоги сформулировать краткое название доклада на тему «почему system design — это плохо». Хочу, чтобы название было броским, смешным и притягивающим взгляд.



"Системный дизайн: Зачем нам эти головоломки?"



# Из чего состоит system design?

- Нужно спроектировать (устно/а может и нет) распределенную систему
- Типовые задачи: twitter, youtube, shortener, chatting service, e-commerce, web crawler, etc.
- Разработчикам эти интервью нравятся

# Что думают нейросети?



- Можно оценить скилы проектирования архитектуры
- Показывает, как человек решает сложные проблемы
- Отражает реальные задачи
- Позволяет сравнивать кандидатов
- Проверяет ваше умение взаимодействовать





Сударь,  
я немного  
приукрасил

Алгоритмические  
собеседования — не очень  
хороший критерий отбора

Сейчас я постараюсь  
показать много  
абсурдных вещей

# Сначала вспомним про $O(n)$

Этот способ очень плох, если мы говорим об алгоритмическом интервью, ведь там мы имеем небольшие наборы данных. В реальном времени квадратичный алгоритм может быть лучше линейного

RE

Помоги сформулировать краткое название доклада. Хочу разносить алгоритмические собеседования. Хочу чтобы название было смешным и очень притягивающим взгляд.



"Алгособесы: как мы разваливаем будущее IT-индустрии и оставляем без работы многообещающих программистов".



# Что такое алгоритмические задачи?

Бесконечный холивар

— Несколько сотен стандартных?

# Список алгоритмов

Один из вариантов

Arrays: Kadane's Algorithm, Floyd's Cycle Detection Algorithm, Knuth-Morris-Pratt Algorithm (KMP), Quick Select Algorithm, Boyer - More Majority Vote Algorithm

Tree: AA Tree, Binary Indexed Tree or Fenwick Tree, Quadtree, Cartesian Tree, Fibonacci heap, Interval Tree, Finger Tree, Crit-bit Trees, Scapegoat Tree, Splay Tree, Suffix Tree, Counted B-Trees, Binary Space Partitioning, Van Emde Boas Tree

Others: Huffman Coding Compression Algorithm, Euclid's Algorithm, Union Find Algorithm, Manacher's Algorithm, Eukerian Path (Hierholzer's Algorithm), Convex Hull | Set 1 (Jarvis's Algorithm or Wrapping), Convex Hull | Set 2 (Graham Scan), Convex Hull using Divide and Conquer Algorithm, Quickhull Algorithm for Convex Hull, Distinct elements in subarray using Mo's Algorithm, Line Sweep Algorithm, MO's Algorithm (Query square root decomposition), Disjoint-set Data Structure, Ackermann Function, Zobrist Hashing, FM-index, Circular buffer, Hungarian Algorithm / Kuhn–Munkres Algorithm / Munkres Assignment Algorithm, Dekker's Algorithm, Winged Edge, Burrows–Wheeler Transform, Zipper, Five Balltree Construction Algorithms, Cuckoo Hashing, Rope (Data Structure), Binary Decision Diagram, Disjoint-set Data Structure, Bloom Filter

# Что такое алгоритмические задачи?

Бесконечный холивар

- Несколько сотен стандартных?
- Несколько из каждой группы?

# Что такое алгоритмические задачи?

Бесконечный холивар

- Несколько сотен стандартных?
- Несколько из каждой группы?
- Какие задачи хорошие? Балансировка скобок? BFS по островам? Строковый калькулятор? Поиск уникального в массиве дублей?

# Что такое алгоритмы?

Алгоритм — последовательность шагов

— Алгоритмы и структуры данных — настолько устоявшаяся фраза, что в общественном сознании эти вещи начинают путаться

— Давайте поищем в интернете и спросим chatgpt, какие алгоритмы правильные

RE

Базовые алгоритмы — это базэврд, по сути. Определи какие алгоритмы являются базовыми для backend + frontend разработчика



Для backend и frontend разработчиков базовыми алгоритмами являются:



1. Сортировка:

- Сортировка пузырьком (Bubble Sort)
- Сортировка выбором (Selection Sort)
- Сортировка вставками (Insertion Sort)
- Быстрая сортировка (Quick Sort)
- Сортировка слиянием (Merge Sort)

2. Поиск:

- Линейный поиск (Linear Search)
- Бинарный поиск (Binary Search)

3. Структуры данных:

- Массивы (Arrays)
- Стек (Stack)
- Очередь (Queue)
- Связанный список (Linked List)
- Дерево (Tree)
- Хеш-таблицы (Hash Tables)

4. Рекурсия:

- Факториал (Factorial)
- Числа Фибоначчи (Fibonacci Numbers)
- Быстрая сортировка (Quick Sort)

5. Динамическое программирование:

- Рюзак (Knapsack)
- Лестница (Staircase)
- Редакционное расстояние (Edit Distance)

6. Графы:

- Обход в глубину (Depth First Search)
- Обход в ширину (Breadth First Search)
- Алгоритм Дейкстры (Dijkstra's Algorithm)
- Алгоритм Прима (Prim's Algorithm)
- Алгоритм Крускала (Kruskal's Algorithm)

RE

## Назови больше базовых алгоритмов



Конечно, вот еще некоторые базовые алгоритмы:



7. Хеш-функции:

- Метод деления с остатком (Division Method)
- Метод умножения (Multiplication Method)

8. Строки:

- Поиск подстроки (Substring Search)
- Конкатенация строк (String Concatenation)
- Алгоритм Кнута-Морриса-Пратта (Knuth-Morris-Pratt Algorithm)

9. Математические алгоритмы:

- Поиск наибольшего общего делителя (Greatest Common Divisor)
- Решето Эратосфена (Sieve of Eratosthenes)
- Быстрое возведение в степень (Exponentiation by Squaring)

10. Алгоритмы оптимизации:

- Методы градиентного спуска (Gradient Descent)
- Методы оптимизации на основе эволюционных алгоритмов (Evolutionary Algorithms)
- Алгоритмы кластеризации (Clustering Algorithms)

11. Алгоритмы машинного обучения:

- Линейная регрессия (Linear Regression)
- Логистическая регрессия (Logistic Regression)
- Метод k-ближайших соседей (K-Nearest Neighbors)
- Деревья решений (Decision Trees)
- Метод опорных векторов (Support Vector Machines)

Как и в предыдущем ответе, это только некоторые из базовых алгоритмов, которые могут быть полезными для разработчиков. В зависимости от области и языка программирования, список может быть дополнен или изменен.

**ВИДЕЛ КАК ТЫ  
ПИШЕШЬ  
QUICKSORT**



**И ТЫ ХОРОШ**

Почему мы решили,  
что алгоритмы на  
собеседованиях —  
это правильно?

**я пытался найти  
первоисточник**

**Не смог! Это просто  
мнение, с которым  
почему-то все  
согласны**

# Но ведь...

- А где же обоснование, данные, доказательства?
- Каждый год пишется несколько статей на тему «как же хороши мои алгоритмы» и каждый раз всё больше голосов, которые с этим не согласны

# Мое мнение

- Это эмпатия
- Апелляция к авторитету: «там умные ребята, лучше знают»/«руководители сказали»/«мой тимлид лучше знает»
- Рационализация: «меня ведь не просто так истязали!»
- Стокгольмский синдром?

**А как часто вы пишете  
«алгоритмы»  
на работе?**



by Getty Images™

by Getty Images™

# Несколько реальных историй

А сколько задач — достаточно, чтобы понять, что «ну норм, берем»

— Коллега написал 3 из 4 решений. Ему отказали

# Несколько реальных историй

А сколько задач — достаточно, чтобы понять, что «ну норм, берем»

- Коллега написал 3 из 4 решений. Ему отказали
- Завалил задачу, написал другую, взяли из-за неё

# Несколько реальных историй

А сколько задач — достаточно, чтобы понять, что «ну норм, берем»

- Коллега написал 3 из 4 решений. Ему отказали
- Завалил задачу, написал другую, взяли из-за неё
- Снизили баллы и не взяли, т.к. очень сенiorный человек не вспомнил кусочек стандартной библиотеки из `itertools`. В фидбэке сказали: «не знаешь `itertools`»

RE

Придумай историю которая дискредитирует алгоритмические собеседования.  
Пример: человек решил тремя способами задачу, а его не взяли, так как он не  
написал еще и четвертый

...

Чтобы разобраться в ситуации, Ким связался с рекрутером компании и попросил обратную связь. Рекрутер объяснил, что компания оценивала не только правильность и эффективность решений, но и скорость выполнения задач. Оказалось, что Киму не удалось решить задачи в ограниченные временные рамки, установленные на собеседовании.

**Нейросеть считает, что решение задач на скорость — это дискредитация алгоритмического подхода к интервью**

 [poteto / hiring-without-whiteboards](#) Public

 Companies that don't have a broken hiring process

 MIT license

 36.3k stars  3.1k forks

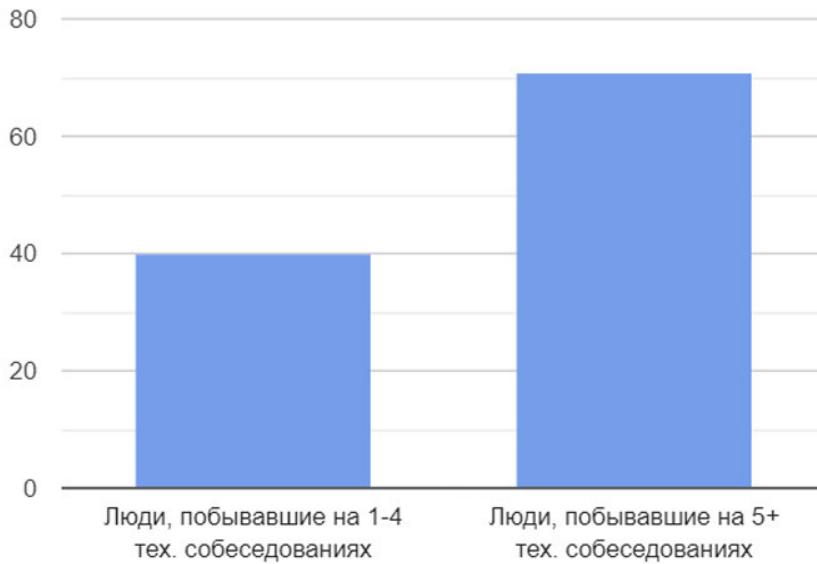
 Starred

 Watch ▾

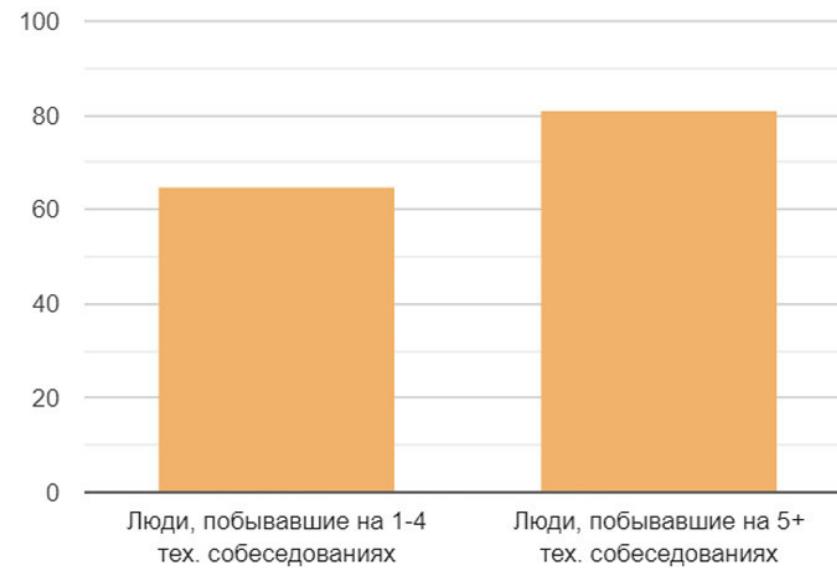
# Абсурдные цифры из реальности

— Люди выяснили, что вероятность быть нанятым на 5 собеседовании в FAANG увеличивается до 60-80 процентов

### **Вероятность прохождения телефонного собеседования в Facebook**



### **Вероятность прохождения телефонного собеседования в Amazon**



# Абсурдные цифры из реальности

- Люди выяснили, что вероятность быть нанятым на 5 собеседований в FAANG увеличивается до 60-80 процентов
- Между собеседованиями «кулдаун» в 6 месяцев (где-то 12)

# Абсурдные цифры из реальности

- Люди выяснили, что вероятность быть нанятым на 5 собеседовании в FAANG увеличивается до 60-80 процентов
- Между собеседованиями «кулдаун» в 6 месяцев (где-то 12)
- Я регулярно в LI читаю посты «наконец решил 1000 задач на литкоде, всего 3 года потратил 😅» 😳 😳 😳

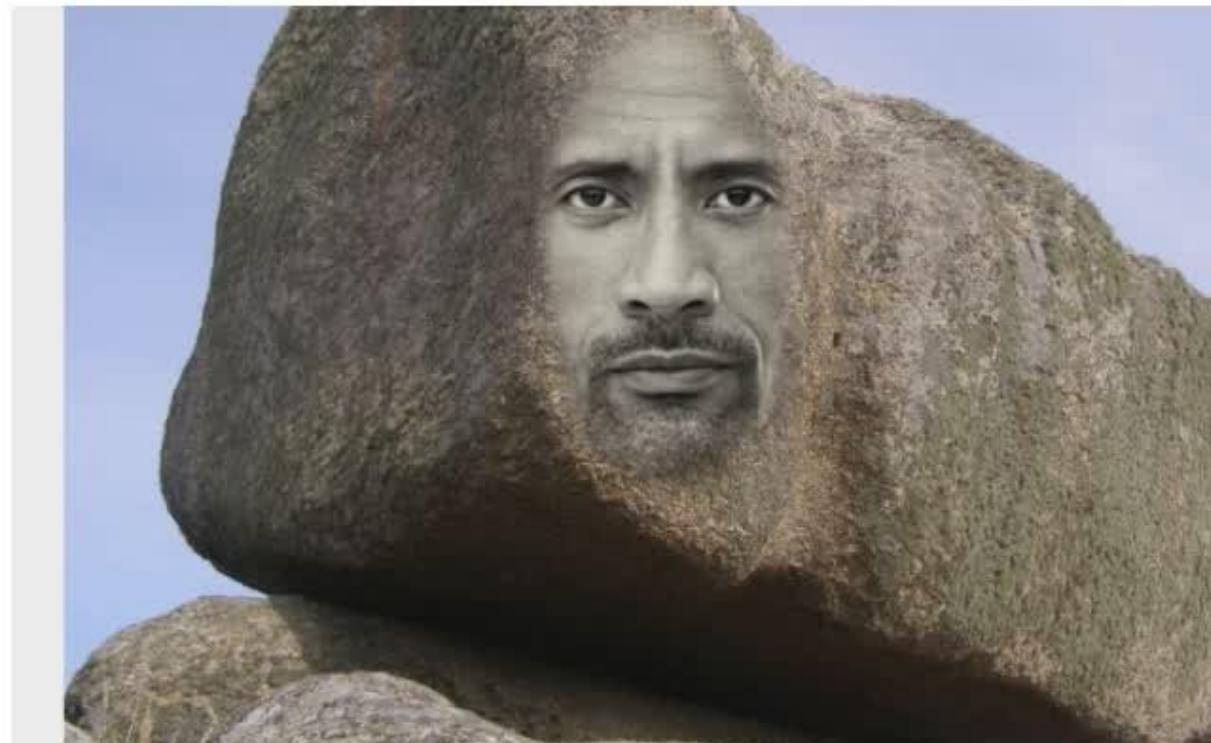
# Абсурдные цифры из реальности

- Люди выяснили, что вероятность быть нанятым на 5 собеседовании в FAANG увеличивается до 60-80 процентов
- Между собеседованиями «кулдаун» в 6 месяцев (где-то 12)
- Я регулярно в LI читаю посты «наконец решил 1000 задач на литкоде, всего 3 года потратил 😅» 😳 😳 😳
- Существует несколько известных мне компаний по подготовке к FAANG-style-интервью и множество отдельных менторов, а также много чатов на тысячи людей

**Наш таймлайн  
проклят! 1!11111**

Time Traveler:drinks water

The timeline:



Есть несколько случаев, когда  
люди проваливали алгоритмические  
задачи и были наняты

И по сей день приносят уйму  
пользы



leetcode-mafia / **cheetah**

Public

Whisper & GPT-based app for crushing remote SWE interviews

CC0-1.0 license

2.8k stars 190 forks



Starred



Watch ▾

**Как обычно говорят про  
собеседования в faang-стиле:  
«нам надо сужать поток, слишком  
много желающих»**

**Наша задача — не  
«резать» поток, а с  
минимальными  
затратами найти коллегу**



**Решил 1000 задач на литкоде?**

**А мог бы эти пять минут  
потратить на de dust**

**«Мы не пустим плохих  
программистов ценой  
отказа нескольким  
хорошим»**

**Как вы поняли, что  
отказали плохим, а  
пустили хороших?**

# «Почему я заваливаю кандидатов на собеседованиях в Google»

Программирование , Карьера в IT-индустрии

Они не дотягивают до нужной планки.



**Это бесчеловечное  
суждение о будущих  
коллегах!**

# Какие проблемы с самим подходом?

— Мы не программируем на виду у незнакомых людей

# Какие проблемы с самим подходом?

- Мы не программируем на виду у незнакомых людей
- Это тренируемый навык, для которого существует целая индустрия и масса книг по подготовке. Он показывает только то, что вы долго тренировались проходить эти интервью

# Какие проблемы с самим подходом?

- Мы не программируем на виду у незнакомых людей
- Это тренируемый навык, для которого существует целая индустрия и масса книг по подготовке. Он показывает только то, что вы долго тренировались проходить эти интервью
- Наша задача — проверять уже существующие навыки у людей, а не заставлять их приобретать новые (к тому же те, которые не влияют напрямую на работу)

# Какие проблемы с самим подходом?

- Мы не программируем на виду у незнакомых людей
- Это тренируемый навык, для которого существует целая индустрия и масса книг по подготовке. Он показывает только то, что вы долго тренировались проходить эти интервью
- Наша задача — проверять уже существующие навыки у людей, а не заставлять их приобретать новые (к тому же те, которые не влияют напрямую на работу)
- Бонус: иногда в качестве задач на 20 минут дают такие сложные проблемы, о которых не так давно ещё писали целые диссертации

# Какие проблемы с самим подходом?

- Мы не программируем на виду у незнакомых людей
- Это тренируемый навык, для которого существует целая индустрия и масса книг по подготовке. Он показывает только то, что вы долго тренировались проходить эти интервью
- Наша задача — проверять уже существующие навыки у людей, а не заставлять их приобретать новые (к тому же те, которые не влияют напрямую на работу)
- Бонус: иногда в качестве задач на 20 минут дают такие сложные проблемы, о которых не так давно ещё писали целые диссертации
- Бонус 2: если к вашему интервью надо готовиться, то это не очень хороший сигнал



А может ли chatgpt покритиковать алгоритмы?

Да! Некоторые аргументы у нас схожи

# System design — неидеален?

# Здесь много схожего с алгоритмами

— Нет правильного ответа. Интерпретация зависит от настроения и взгляда интервьюера полностью. А «продают» нам это как объективную оценку

**ОЧЕНЬ ПОНРАВИЛОСЬ**

**ПОКУПАЮ!**

# Здесь много схожего с алгоритмами

- Нет правильного ответа. Интерпретация зависит от настроения и взгляда интервьюера полностью. А «продают» нам это как объективную оценку
- Та же нетипичная ситуация. Единицы проектируют архитектуру с незнакомыми и, возможно, недружелюбными людьми

# Здесь много схожего с алгоритмами

- Нет правильного ответа. Интерпретация зависит от настроения и взгляда интервьюера полностью. А «продают» нам это как объективную оценку
- Та же нетипичная ситуация. Единицы проектируют архитектуру с незнакомыми и, возможно, недружелюбными людьми
- Стандартный набор, который формулируется в *system design primer* и *cracking system design interview*

# Здесь много схожего с алгоритмами

- Нет правильного ответа. Интерпретация зависит от настроения и взгляда интервьюера полностью. А «продают» нам это как объективную оценку
- Та же нетипичная ситуация. Единицы проектируют архитектуру с незнакомыми и, возможно, недружелюбными людьми
- Стандартный набор, который формулируется в *system design primer* и *cracking system design interview*
- Компании уже поняли, что все расписано в деталях и усложняют этот процесс. А где же действие «остановиться и подумать»?

# Здесь много схожего с алгоритмами

- Нет правильного ответа. Интерпретация зависит от настроения и взгляда интервьюера полностью. А «продают» нам это как объективную оценку
- Та же нетипичная ситуация. Единицы проектируют архитектуру с незнакомыми и, возможно, недружелюбными людьми
- Стандартный набор, который формулируется в *system design primer* и *cracking system design interview*
- Компании уже поняли, что все расписано в деталях и усложняют этот процесс. А где же действие «остановиться и подумать»?
- Вы правда верите в масштабируемый твиттер за полчаса на коленке?

# Общие проблемы

# Почему подходы дают искажённую картину

- Умение проходить system design и алгоритмические секции — это очень специальный, вырабатываемый навык, наличие которого говорит только о вашем умении проходить system design и алгоритмические секции, а не о том как вы программируете в обычной ситуации
- Обычные навыки программиста: усидчивость, упорство, умение работать в команде
- Не вдаваясь в левую повестку, мы сужаем разнообразие, замыкаемся в своем «правильном» (тм) мнении
- Посмотрите на тех, кто проходил такие интервью: они всегда за них или как минимум не против



F(O)CUSED  
COLLECTION

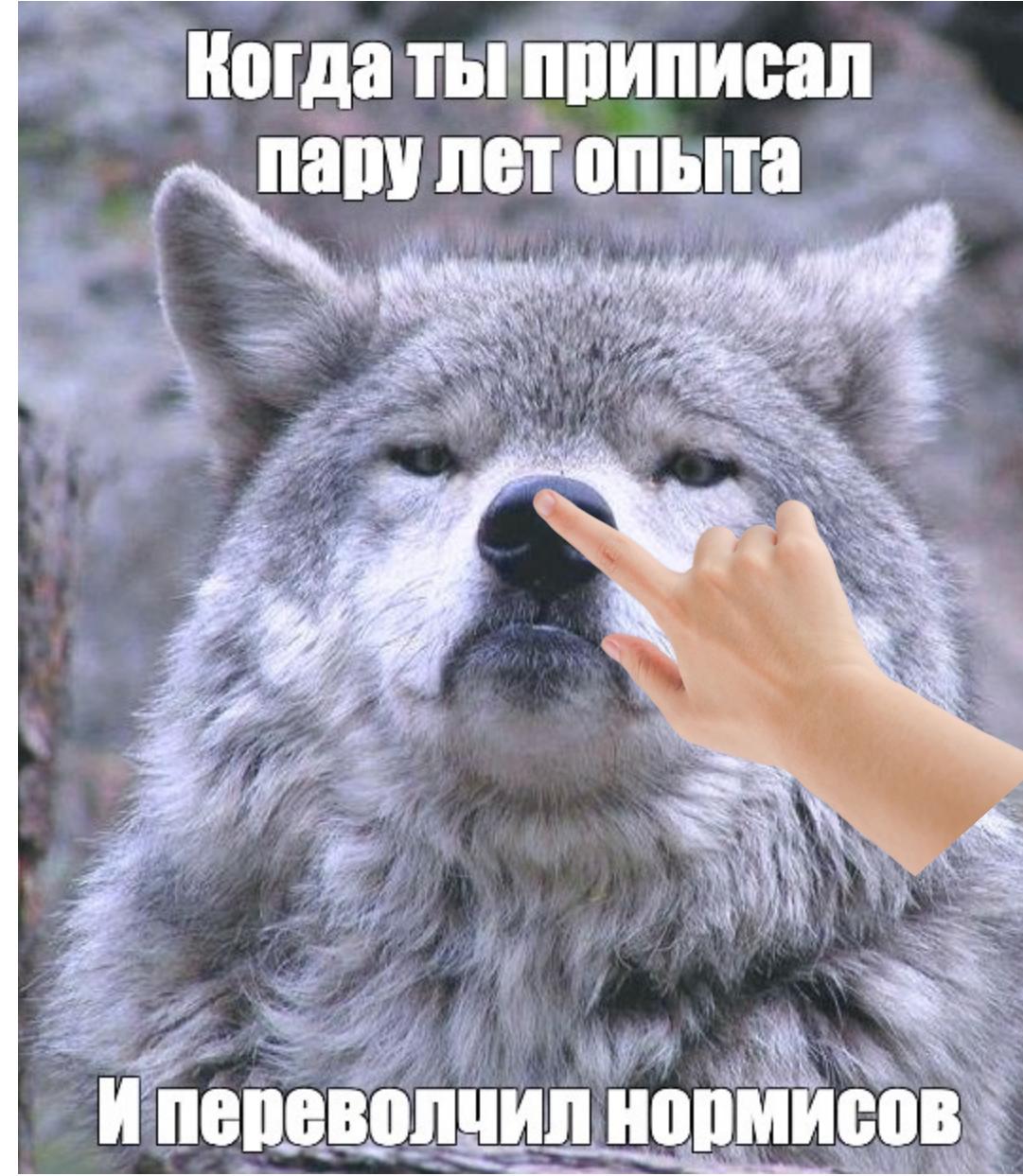
ID: 465125408  
FOCUSDECOLLECTION.COM

# Ещё больше проблем

# Коротко

— Годы опыта

**Когда ты приписал  
пару лет опыта**



**И переволчил нормисов**

# Коротко

- Годы опыта
- Тестовые задания



**МЫ ВАМ**

**ПЕРЕЗВОНИМ**

# Коротко

- Годы опыта
- Тестовые задания
- Низкая эффективность огромных конвейеров отбора

# Коротко

- Годы опыта
- Тестовые задания
- Низкая эффективность огромных конвейеров отбора
- Цена ошибки найма не так велика, как это показывают

# Коротко

- Годы опыта
- Тестовые задания
- Низкая эффективность огромных конвейеров отбора
- Цена ошибки найма не так велика как это показывают
- Многие люди получают психологический урон

# Что делать?



**Все мои рецепты будут  
очень банальными и  
очень субъективными**

# BREATHING



# WILL HELP YOU LIVE

**Просто поговорить с  
кандидатами**

# Субъективность — страшно?

Я так не думаю

- Мы люди, мы всегда субъективны
- Если мы боимся субъективности, стараемся сделать процесс «бесчеловечным» и от этого он делает людям больно, и субъективность только увеличивается

# Как исправить?

Принять субъективность и извлечь пользу

- Принять свою субъективность. В бонус можно извлекать пользу индивидуальным подходом, харизмой нанимающих
- Создать общий скелет интервью
- Делать артефакты
- Искать совместимых с нами людей. Если мы совместимы с организацией, то совместимые с нами люди совместимы с организацией тоже

# Чем заменить алгоритмы?

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

— Одним вдумчивым интервью

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

- Одним вдумчивым интервью
- Спрашивать не quiz-вопросы и не стандартные вопросы из «100 лучших вопросов по python», а действительно просить людей думать

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

- Одним вдумчивым интервью
- Спрашивать не quiz-вопросы и не стандартные вопросы из «100 лучших вопросов по python», а действительно просить людей думать
- Например: вместо mutable vs. immutable можно поговорить о типизации

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

- Одним вдумчивым интервью
- Спрашивать не quiz-вопросы и не стандартные вопросы из «100 лучших вопросов по python», а действительно просить людей думать
- Например: вместо mutable vs. immutable можно поговорить о типизации
- Например: вместо «что такое итератор?» спросить iterable vs iterator и покопаться в протоколе

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

- Одним вдумчивым интервью
- Спрашивать не quiz-вопросы и не стандартные вопросы из «100 лучших вопросов по python», а действительно просить людей думать
- Например: вместо mutable vs. immutable можно поговорить о типизации
- Например: вместо «что такое итератор?» спросить iterable vs iterator и покопаться в протоколе
- Важная мысль: иметь свое уникальное интервью, с важными именно для вас и вашей компании вопросами

# Чем заменить алгоритмы?

Я вижу это так, но вам я бы предложил подумать что бы хотели спрашивать людей именно вы

- Одним вдумчивым интервью
- Спрашивать не quiz-вопросы и не стандартные вопросы из «100 лучших вопросов по python», а действительно просить людей думать
- Например: вместо mutable vs. immutable можно поговорить о типизации
- Например: вместо «что такое итератор?» спросить iterable vs iterator и покопаться в протоколе
- Важная мысль: иметь свое уникальное интервью, с важными именно для вас и вашей компании вопросами
- А ещё можно попросить поревьюить PR и послушать аргументы. Эта практика может быть полезна и для понимания архитектурного бэкграунда

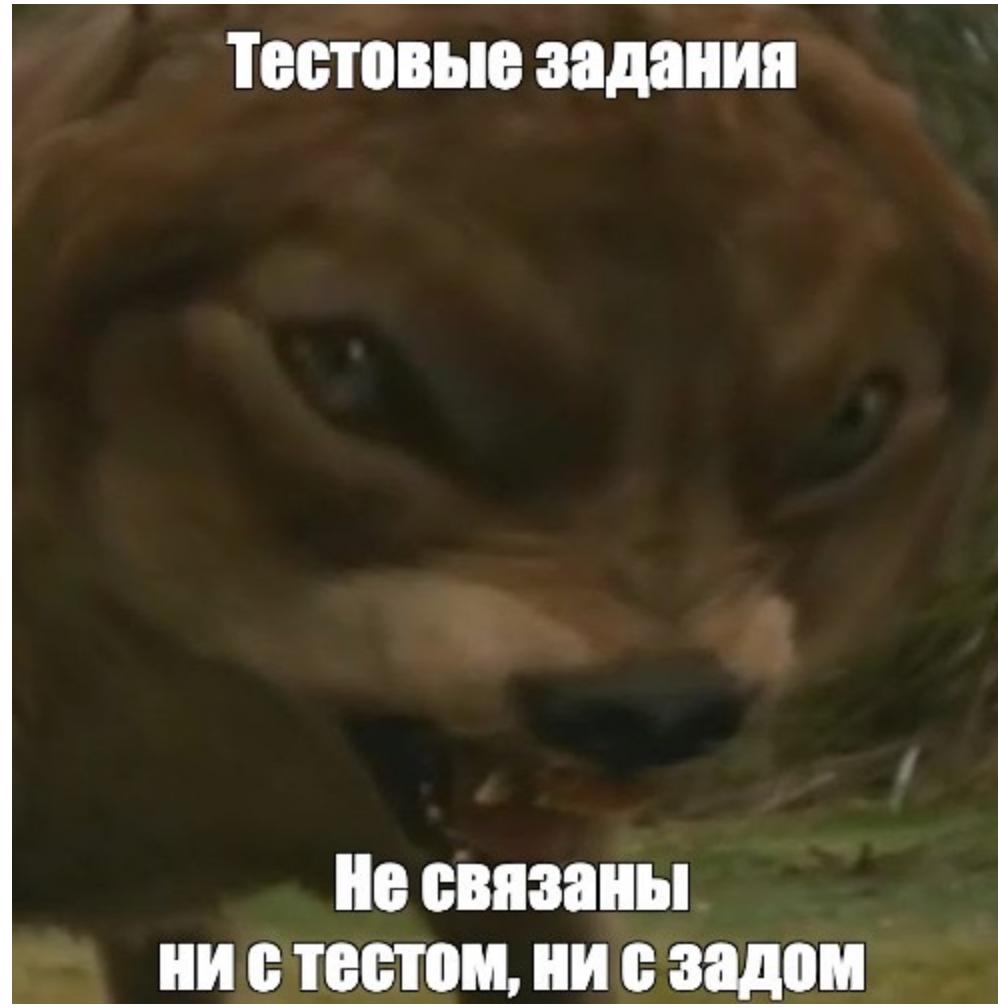
# Чем заменить system design?

# Чем заменить system design?

Например:

- Поговорить о проблемах распределенных систем: CAP, консенсус, split brain, транзакции, гонки, локи, дедлеки
- Поговорить о типовых паттернах распределенных систем: db per service, event driven, saga, cqrs и т.п.
- Обсудить архитектурные проблемы, возникающие в процессе долгой жизни вашей системы. Например, как поддерживать и развивать целостную архитектуру?
- Поговорить о типовых проблемах messaging driven, event driven систем, послушать кандидата, как он решает такие проблемы
- Обсудить reactive manifest
- Обсудить масштабирование

# Чем заменить тестовые?



**Тестовые задания**

**Не связаны  
ни с тестом, ни с задом**

**Ничем!**

# Как сравнивать кандидатов?

# А это точно помогает вам и бизнесу?

- Сравнение кандидатов действительно цель?
- У вас есть хороший кандидат, который вам подходит, что вам ещё нужно?
- Описанные способы все субъективны или суперсубъективны. Значит, ни о каком объективном сравнении мы не можем говорить изначально!
- Вы нанимаете первого кандидата, который вас устраивает и всё

# А это точно помогает вам и бизнесу?

- Сравнение кандидатов действительно цель?
- У вас есть хороший кандидат, который вам подходит, что вам ещё нужно?
- Описанные способы все субъективны или суперсубъективны. Значит, ни о каком объективном сравнении мы не можем говорить изначально!
- Вы нанимаете первого кандидата, который вас устраивает и всё

Или

- Вспоминаете задачу о разборчивой невесте

**Неплохая практика для  
выравнивания —  
записывать ваши  
впечатления**

Форма сохраняет состояние между перезагрузками страницы. По окончании заполнения достаточно нажать «распечатать» в конце страницы или + . PDF принтер сохранит получившийся отчёт.

**И[О]Ф\***

Денис Аникин

**Решение\***

- Рекомендую    Не рекомендую    Стоит попробовать в другие команды

Рекомендую, если

**Уровень\***

- Trainee    Junior    Junior+    Middle-    Middle    Middle+    Senior-    Senior    Senior+

**Рекомендации (для ОС кандидату)**

Может быть полезно улучшить знания в области асинхронного программирования, так же распределенных архитектур и особенно алгоритмов консенсуса, таких как [Paxos](#) или [RAFT](#)

**Софт скилы**

Кандидат коммуникабельный, однако довольно много говорит

**Комментарии**

Рекомендую попробовать в команду где много общительных людей

**Дополнительные артефакты**

Для лайвкодинга: <http://collabedit.com/new>

**Python**

<https://docs.python.org/3/library/index.html>  
<https://docs.python.org/3/reference/index.html>

Frameworks

- Нет ответа    django    flask    aiohttp    fastapi    sanic    litestar    blacksheep  
 socketify.py    robyn

# Исходник «интервьюилки»

<https://github.com/xfenix/interview-engine>



Что, если я найму  
«плохого» кандидата?



**Ничего страшного**

**Ответ на почти все вопросы  
даёт ТК:  
испытательный срок!**

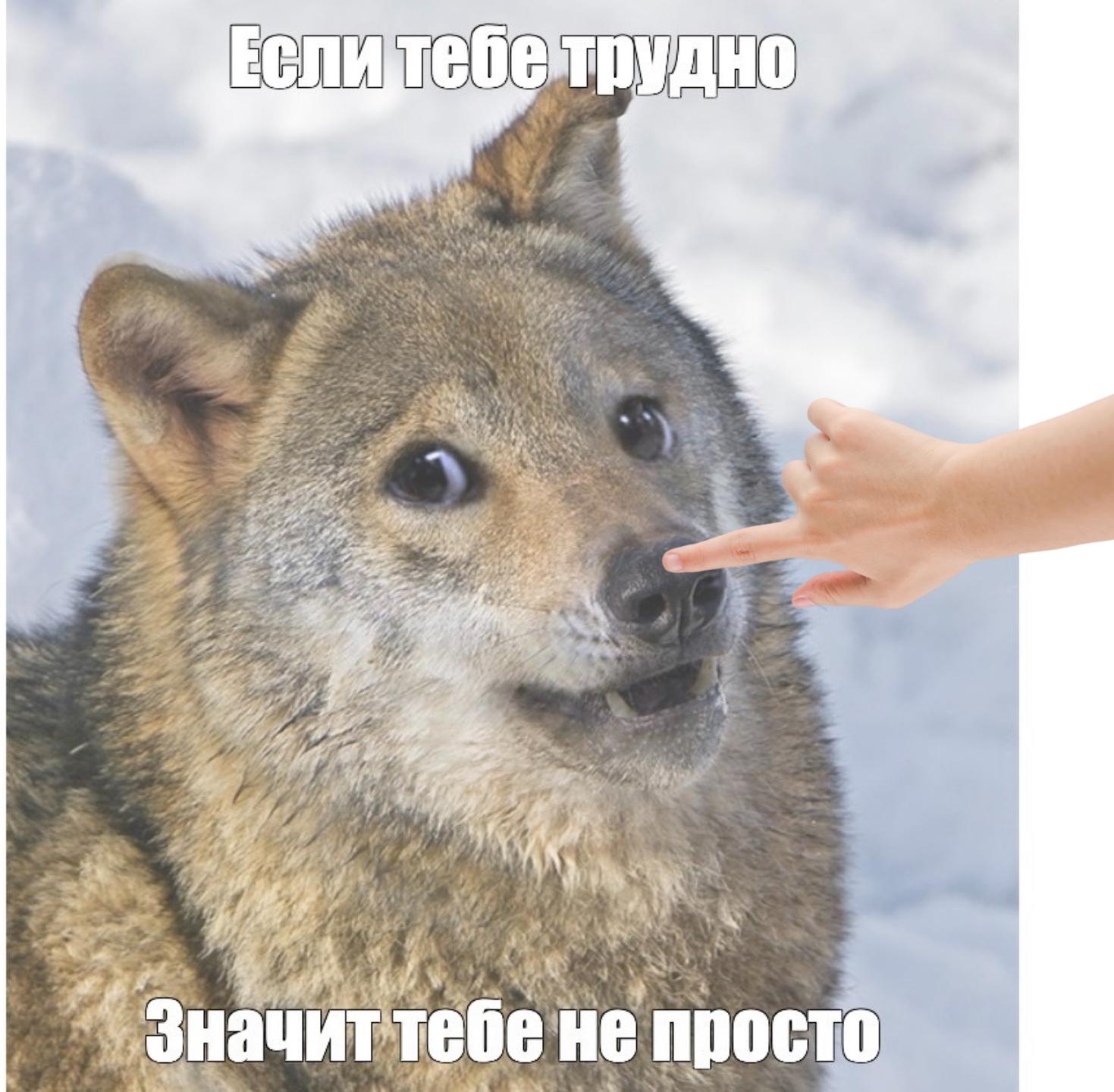
**Что если наняли  
мидла, а он  
притворяющийся  
джуниор?**

**Если он делает задачи мидла —  
он мидл.**

**Если не делает, есть  
испытательный срок**

**Все так же, как и со всеми  
сотрудниками**

**Если тебе трудно**



**Значит тебе не просто**

**«Наняли тяп-ляпателя на фреймворках, джанго-программиста, теперь алгоритмам не научить»**

**А вот наняли бы алго-ниндзя, он бы любую технологию осилил!**

**Так ли это на самом деле?  
Где доказательства?**

Как нам понять, что  
candidate\_name  
качественно и быстро  
программирует?



**Внимательно изучите  
резюме, посмотрите гитхаб  
(при наличии), поговорите с  
человеком**

**Скоринг, баллы, веса —  
ВОТ ОТВЕТ!**

**Точно так же субъективно:  
сколько баллов стоит  
хороший сеньор, а сколько  
плохой мидл?**

# Что ещё хочется отметить важного

— Перед нами опытный специалист, и мы с ним равны

# Что ещё хочется отметить важного

- Перед нами опытный специалист, и мы с ним равны
- Наше интервью — диалог, а не допрос

# Что ещё хочется отметить важного

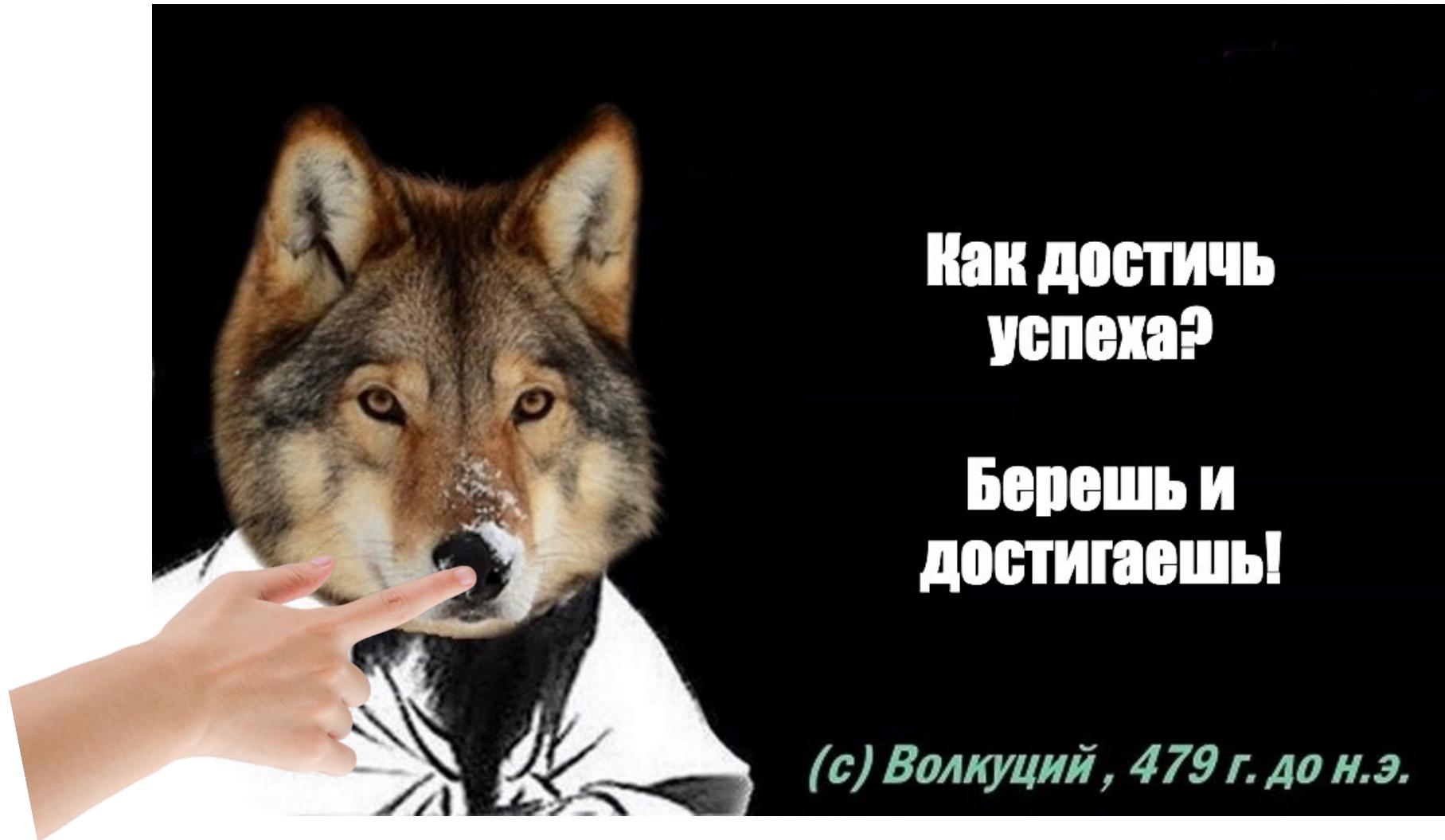
- Перед нами опытный специалист, и мы с ним равны
- Наше интервью — диалог, а не допрос
- В идеале, если человек уходит от нас с восклицанием «спасибо, это было полезно/интересно», а не с психологической травмой

# Что ещё хочется отметить важного

- Перед нами опытный специалист, и мы с ним равны
- Наше интервью — диалог, а не допрос
- В идеале, если человек уходит от нас с восклицанием «спасибо», это было полезно/интересно», а не с психологической травмой
- Не стоит стесняться принимать обратную связь от кандидатов и менять что-то, исходя из неё

# Что ещё хочется отметить важного

- Перед нами опытный специалист, и мы с ним равны
- Наше интервью — диалог, а не допрос
- В идеале, если человек уходит от нас с восклицанием «спасибо», это было полезно/интересно», а не с психологической травмой
- Не стоит стесняться принимать обратную связь от кандидатов и менять что-то, исходя из неё
- Лучше всего не пытаться оценить человека, а ответить на вопрос: сможем ли мы работать с этим человеком?



**Как достичь  
успеха?**

**Берешь и  
достигаешь!**

*(с) Волкуций, 479 г. до н.э.*

# Полезные ссылки

- <https://www.techinterviewhandbook.org/> выдающийся материал для подготовки
- <https://github.com/donnemartin/system-design-primer>
- <https://github.com/poteto/hiring-without-whiteboards>



# Спасибо



<https://xfenix.ru>

<https://github.com/xfenix>

<https://www.linkedin.com/in/xfenix/>

