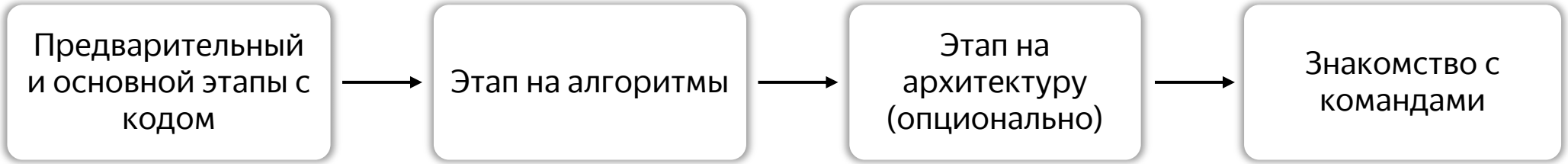
Яндекс Go

Подготовка к интервью в Технологическую платформу

После успешного прохождения предварительного интервью с рекрутером мы приглашаем вас на следующие этапы по Zoom:



- Предварительный и основной этапы с кодом содержат всего 4-8 задач на кодинг в онлайн-редакторе. Также вас могут спросить об опыте и работе с разными технологиями (ЯП, ОС, БД). Длительность каждого этапа – 1 час.
- Алгоритмический этап состоит из 2 задач на алгоритмы в онлайн-редакторе, важно решить обе. Длительность этапа – 1 час 15 мин.
- Архитектурный этап содержит задачу, для решения которой нужно отрисовать архитектуру сервиса в редакторе. Длительность – 1 час.
- Собеседования проходят по Zoom, подключайтесь с ноутбука или компьютера, не забудьте включить звук и камеру.
- По итогам интервью мы свяжется с вами в течение 2-3 дней.

Яндекс Go

1 Во время подготовки к интервью

- Потренируйтесь решать задачи на одном из ресурсов самостоятельно, без подсказок, любым известным вам способом.
- При решении задач всегда объясняйте себе свое решение: почему именно такое, можно ли эффективней? Если нет, то почему?
- Подумайте про все граничные условия, где каждое из решений должно переходить в тесты.
- Потренируйтесь решать код без IDE, так как на интервью у вас не будет возможности ее использовать.
- После того, как вы решите задачу, посмотрите обсуждение задачи (например, в Discuss на Leetcode), чтобы убедиться, что ваше решение оптимальное.

2 На интервью

- После того, как интервьюер озвучил условия, обязательно расскажите ему задачу своими словами и убедитесь, что вы правильно поняли условия.
- Не бойтесь задавать вопросы по задаче: какие данные, какие допущения можно сделать и прочие вопросы, которые помогут понять, какое решение от вас ждут.
- До написания кода проговорите свое решение, обсудите его правильность с интервьюером.
- Не старайтесь написать сразу идеальное решение – начните с базового (набросайте скелет), посмотрите как решение будет работать на разных данных. Если нужно, уточните мнение интервьюера.
- Умение объяснять важно в командной работе, поэтому вам нужно будет объяснить свое решение интервьюеру. Если есть необходимость, подумайте заранее, как вы будете это делать.

Что посмотреть для подготовки:

Как решать алгоритмические секции:

- Разбор задач на youtube: простые задачи и более сложные.
- Статья «Как проходят алгоритмические секции на собеседованиях в Яндекс».
- Общие советы для подготовки.
- Видеолекции курса «Алгоритмы и структуры данных».
- Учебник «Алгоритмы» С. Дасгупта, Х. Пападимитриу, У. Вазирани

- Контест для подготовки.
- Leetcode (задачи уровня easy и medium с тегами: Array, String, Tree, Binary Search, Hash table, Depth-first Search, Breadth-first Search, Two Pointers, Stack, Backtracking; задачи с разным уровнем acceptance).
- Geekforgeeks (задачи от уровня medium table, Depth-first Search, Breadth-first Search, Two Pointers, Stack, Backtracking; задачи с разным уровнем acceptance).

Примеры задач:

Задача AA1

Задача AA2

Яндекс Go

Для подготовки к Архитектуре

При проектировании сервисов и систем важно обращать внимание на:

- Сервис нужно проектировать как систему, состоящую из гибких, независимых частей мало влияющих друг на друга. Например, выбор БД в качестве API должен быть как минимум серьезно обоснован, поскольку пронизывает всю систему и будет влиять почти на все в ней.
- Выбор технологий для реализации должен быть обоснованным. Нельзя выбирать незнакомые или малознакомые технологии просто потому, что их кто-то уже использовал. Возможно, они не будут отвечать нашим требованиям.
- Проектируя любую часть системы нужно иметь ввиду, что то, что в принципе может сломаться, обязательно когда-нибудь сломается. Следует минимизировать ситуации, в которых систему придется поднимать руками.
- Сервис, для которого требуется обеспечить неопределенный уровень масштабируемости изначально должен проектироваться с расчетом на то, что когда-нибудь мощностей одной машины нам перестанет хватать. Нельзя рассчитывать, что нам «хватит одной машины».
- Система не должна иметь одну точку отказа (например, одна БД, использующая Postgres foreign data wrappers в качестве гейтвея для доступа к данным на других серверах).

Для эффективного решения дизайнерских задач в области инфраструктуры важно хорошо понимать:

- Принципы работы систем очередей вроде RabbitMQ (нюансы их работы, их основные отличия от реляционных СУБД, гарантии персистентности и средства обеспечения отказоустойчивости).
- Механизмы синхронизации в условиях многопоточного доступа к одним и тем же данным в БД (например, в общем случае нельзя использовать уровни изоляции транзакций для установки блокировок в БД).
- Гарантии персистентности для основных хранилищ (реляционные СУБД, например) и условия при которых они предоставляются.
- Как надежно решаются типовые задачи, используемые в большом количестве систем (например, очереди в Redis или БД).
- Как собирать метрики таким образом, чтобы по ним можно было определить, какая именно проблема и в каком месте системы возникла. Нужно метрить все важные процессы, которые в системе происходят и понимать, какие именно метрики для них являются стандартными и какие отклонения свидетельствуют о наличии проблем.

Яндекс Go

Яндекс Go