Multitrack

Памятка для подготовки к секциям для бэкенд-разработчиков

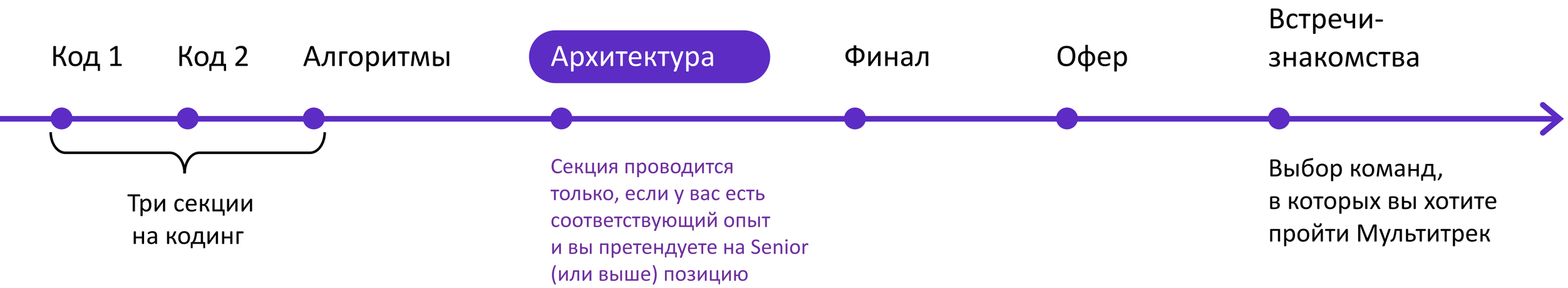
Я team

Привет, герой Мультитрека!

Для подготовки к секциям рекомендуется ознакомиться с общим гайдом Яндекса

Данная памятка носит дополняющий характер
и содержит советы, которые могут быть полезны
именно героям Мультитрека

Весь трек


Код 1
Код 2
Алгоритмы
Архитектура
Финал
Офер
Встречи-
знакомства
Три секции
на кодинг
Секция проводится
только, если у вас есть
соответствующий опыт
и вы претендуете на Senior
(или выше) позицию
Выбор команд,
в которых вы хотите
пройти Мультитрек

Какие этапы будут

На секциях с кодом проверяем знание вашего основного языка программирования, знание предметной области

Вас также могут спросить об опыте и работе с разными технологиями (ЯП, ОС, БД).

Алгоритмы проводим без привязки к языку программирования, проверяем инженерные навыки, работу с алгоритмами и структурами данных

На секции важно посчитать асимптотики, оценить сложность и правильность алгоритма

Секция проводится как System Design

Без кода, без задачек

Я team

Памятка по техническим этапам

Код и Алгоритмы

Знать структуры данных
и сложность O(?) операций
над ними

Знать концепты, действия над ними
и для чего они используются

Знать, как структуры и концепты
реализуются в выбранном языке
программирования

Шаблон стандартной задачи из секции с кодом:

Дан список {строк, символов, чисел, бананов, кактусов}
(или списки, не больше двух). Проанализировать его и:

- Найти определенные элементы согласно критерию
- Перегруппировать элементы или вернуть подгруппу
- Суммаризовать и дать ответ на вопрос «сколько?»
  или «есть ли?»

Для этого:

- Использовать структуры данных {массив, словарь, множество,
  дополнительные классы объектов} порой сразу несколько
- Возможно использование концептов и действий над ними.
  Знание концептов помогает выбрать правильную модель для
  решения задачи

Как правило, но не обязательно:

- Траверсировать для предварительного анализа,
  выполняя хитрые действия в процессе
- Траверсировать для окончательного анализа, выполняя хитрые
  действия в процессе.

Памятка по техническим этапам

Код и Алгоритмы

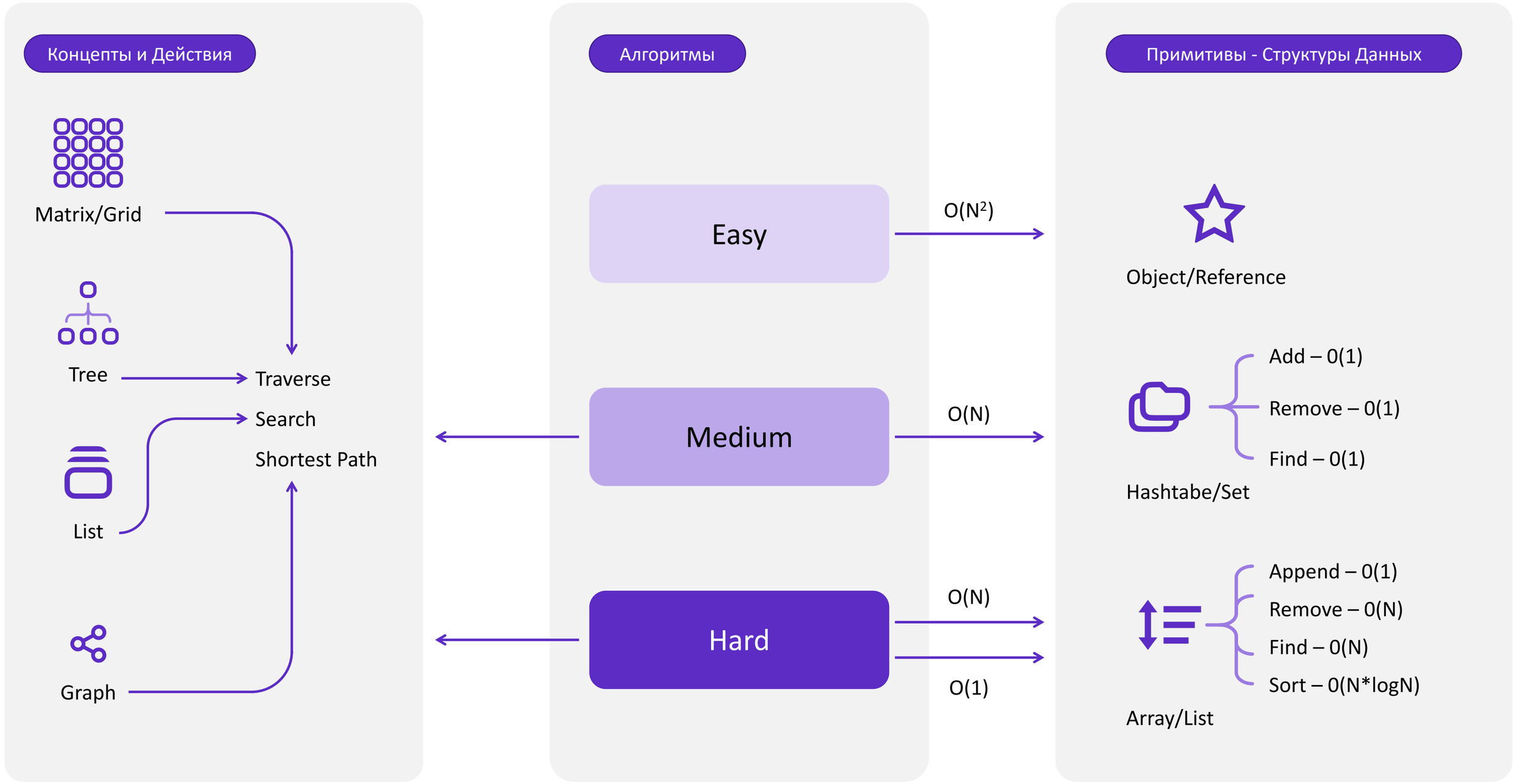
Важно уметь запускать/дебажить код в голове и не забывать про corner cases

Важно, чтобы алгоритм был максимально эффективным — как правило O(N)

Для практики подойдут любые средние задачи из leetcode.com, которые соответствуют данному выше шаблону.

На них можно отработать:

- План проработки алгоритма: Какие структуры я буду использовать чтобы получилось O(N) эффективно? Как правило структуры с O(1) временем доступа + один или два прохода по исходному списку
- Запуск и отладку алгоритма в голове
- Проговаривание вслух алгоритма, чтобы было понятно бабушке. Не секрет, что сначала надо проговорить решение вслух и только потом писать код


Концепты и Действия
Алгоритмы
Примитивы - Структуры Данных
Matrix/Grid
Tree
Traverse
Search
Shortest Path
List
Graph
Easy
Medium
Hard
O(N²)
O(N)
O(N)
O(1)
Object/Reference
Add – 0(1)
Remove – 0(1)
Find – 0(1)
Hashtabe/Set
Append – 0(1)
Remove – 0(N)
Find – 0(N)
Sort – 0(N*logN)
Array/List

Памятка по архитектуре

Про секцию

Предстоит определить и сформулировать функциональные и технические требования, спроектировать высокоуровневую архитектуру, детально описать один из компонентов, оценить вычислительные ресурсы, необходимые для полномасштабного внедрения.

Пример

Спроектировать сервис Яндекс Такси.
Как правило, задача формулируется в очень общих чертах, поэтому здесь важна активная коммуникация: задавайте уточняющие вопросы, выдвигайте обоснованные предположения и идеи, аргументируйте свою точку зрения.

Советы при подготовке

Обоснуйте выбор технологий.

Не рекомендуется брать незнакомые или малознакомые технологии просто потому, что их кто-то где-то использовал. Возможно, они не будут отвечать требованиям задачи.

Любой сервис рано или поздно приходится масштабировать, проектируйте с учётом того, что когда-нибудь мощностей одной машины и одной базы нам перестанет хватать.

Помните: то, что может сломаться, обязательно сломается, нужно минимизировать ситуации, в которых систему придется «поднимать руками». Также стоит предложить набор метрики мониторингов, которые помогут понять, насколько стабильно работает система.

Система должна иметь как можно меньшее количество точек отказа. Например, единственная инсталляция монолитного бэкенда из-за проблем с сетью, электричеством, перегревом в дата-центре приведёт к недоступности всего сервиса.

Сервисы спроектируйте как систему, состоящую из гибких, независимых частей, которые мало влияют друг на друга. Например, использование одной БД одновременно из нескольких частей должно быть серьёзно обосновано, поскольку изменение её схемы для одной части будет влиять на другие.

Что повторить

- Принципы хранения данных в реляционных базах вроде PostgreSQL и нереляционных вроде MongoDB. Понимать, почему стоит разделять хранение часто и редко меняющихся данных.
- Как собирать метрики, чтобы определить проблемы и место их возникновения. Какие есть средства для их агрегации настройки мониторингов.
- Принципы работы систем очередей вроде RabbitMQ, брокеров сообщений вроде Apache Kafka. Важно понимать, какие гарантии они предоставляют, как обеспечивается отказоустойчивость.
- 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 и условия, при которых они предоставляются. Насколько надежно решаются типовые задачи с использованием этих хранилищ.

Памятка по финалу

Здесь мы подробнее узнаем о вас и расскажем о себе. Финал всегда проходит по формату обсуждения ваших навыков и разбора кейсов из реальной работы.

Как подготовиться:

Заранее подумайте о проекте, в реализации которого вы играли ключевую роль. Важно выбрать проект, актуальный вашей позиции и из недавнего прошлого, чтобы вы смогли назвать все детали. Нам будет важно понять, за что у вас была ответственность в этом проекте, и какие задачи стояли перед вами.

Материалы для подготовки

Площадки для тренировки

CodeRun | Leetcode | Hackerrank

ⓘ Обязательно при подготовке

Для удобства можете воспользоваться готовым списком:

binary search:

- https://leetcode.com/problems/binary-search/
- https://leetcode.com/problems/search-in-rotated-sorted-array/

hash table:

- https://leetcode.com/problems/single-number/
- https://leetcode.com/problems/two-sum/

queue/stack:

- https://leetcode.com/problems/valid-parentheses/

dfs/bfs:

- https://leetcode.com/problems/number-of-islands/

sort:

- https://leetcode.com/problems/merge-intervals/

two pointers:

- https://leetcode.com/problems/container-with-most-water/
- https://leetcode.com/problems/partition-labels/

sliding window:

- https://leetcode.com/problems/sliding-window-median/
- https://leetcode.com/problems/sliding-window-maximum/
- https://leetcode.com/problems/longest-repeating-character-replacement/

tree:

- https://leetcode.com/problems/same-tree/
- https://leetcode.com/problems/symmetric-tree/

Материалы для подготовки

Статьи и книги

Опционально, по-вашему желанию

Алгоритмы

- Собеседования в Яндекс: взгляд бэкендера
- «Асимптотический анализ алгоритмов»
- Оценка сложности алгоритмов
- Подборка по алгоритмам

Архитектурная секция

- Как проходят архитектурные секции собеседования в Яндексе
- Опыт разработчиков
- Гайд про архитектуру
- System Design interview
- Числа, которые точно нужно знать

Материалы для подготовки

Видео

Опционально, по-вашему желанию

Алгоритмы

- Курс «Подготовка к алгоритмическому собеседованию»
- Разбор задач на YouTube: простые задачи и более сложные;
- Вебинар на YouTube «Открытое алгоритмическое собеседование»

Архитектурная секция

- Запись стрима «Как проектировать архитектуру сервиса»
- Coding and System Design Interview Questions

Для вашего саморазвития

- System Design for Beginners
- Intro to Architecture and Systems Design Interviews
- Лекция про масштабируемость, Дэвид Малан

Multitrack

Отличной подготовки и удачи!