

Советы для подготовки к алгоритмическим собеседованиям (для всех уровней разработчиков)

Автор: Артём Шумейко
([YouTube](#), [tg-канал](#), [tg-сообщество](#))

Предисловие

Всем привет, меня зовут Артём. Я автор YouTube канала о Python разработке, а также телеграм-сообщества для Python разработчиков.

Я прошел все этапы собеседований (3 созвона с HR и командой, 3 алгоритмических собеседования) в Яндекс на позицию backend разработчика и **получил оффер** на позицию Middle разработчика.

Я также прошел большое количество собеседований в корпорациях и стартапах и часто решал алгоритмические задачи на лайв-кодинге, поэтому я имею представление о том, какие темы нужно знать в зависимости от вашего уровня и компании, в которую вы подаетесь.

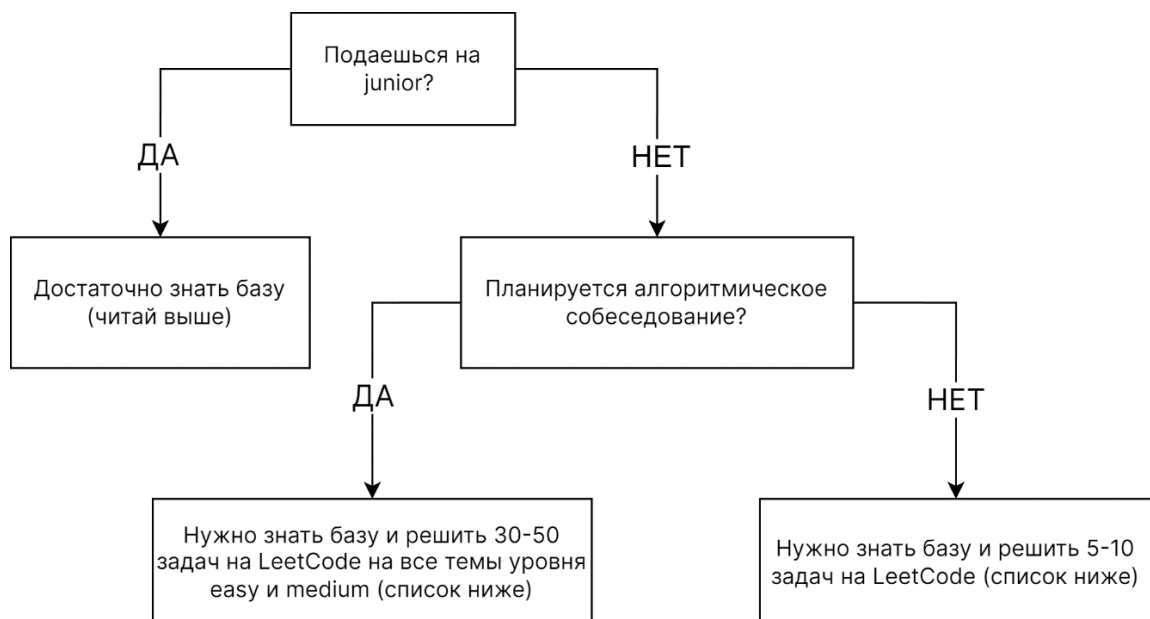
База

Есть темы, которые должны знать все разработчики. Сюда входит:

- "O" большое - оценка сложности по времени и памяти
- Все структуры данных (list, tuple, dict, set) их (не)изменяемость
- Время сортировки, поиска, удаления элементов для каждой структуры данных (ответы будут ниже)

Схема

Схема ниже позволяет определить, что вам нужно сделать для подготовки к собеседованию в части алгоритмов.



О-нотация

В этом разделе представлена информация по оценке сложности действий со структурами данных, которые часто спрашивают на скринингах и собеседованиях.

Сложность по времени	Структура данных			
Операция	list (список)	tuple (кортеж)	dict (словарь)	set (сет)
Обращение по индексу/ключу	O(1)	O(1)	O(1)	-
Добавление элемента в конец	O(1)	-	-	-
Добавление элемента в середину	O(n)	-	O(1)*	O(1)*
Удаление (в среднем)	O(n)	-	O(1)	O(1)
Удаление последнего элемента	O(1)	-	-	-
Удаление первого	O(n)	-	-	-
Поиск элемента (x in y)	O(n)	O(n)	O(1)	O(1)
Нахождение мин./макс. значения	O(n)	O(n)	O(n)	O(n)
Итерация по всем элементам	O(n)	O(n)	O(n)	O(n)
Сортировка элементов	O(n log n)	O(n log n)	O(n log n)	O(n log n)

*в dict и set нет начала, середины и конца

В идеале, конечно, еще знать про [deque](#) и [heap](#) (кучу), но про них редко спрашивают.

LeetCode задачи





Если вы хотите стать **junior** разработчиком, изредка на собеседованиях вас могут попросить решить простенькую задачу на алгоритмы. Вот список популярных алгоритмических задач:

1. [Two sum](#) – классика, 1ая задача на LeetCode, спрашивают очень часто
2. [Валидные скобки](#) – также очень часто спрашивается
3. [Бинарный поиск](#) – познакомит вас с бинарным поиском
4. [Дубликат в списке](#) – база, учит пользоваться хэш-таблицей
5. [Как заработать котлету на рынке](#) – классическая задача на скользящее окно
6. [Макс. строка без повторяющихся символов](#) – простые манипуляции со строками
7. [Макс. общий префикс](#) – опять же работа со строками
8. [Римское число в арабское](#) – одна из моих любимых задач на строки и хэш-таблицы
9. [Число-палиндром](#) – научит базовым манипуляциям с целочисленным делением и остатками
10. [Подвинуть нули](#) – научит работать со скользящим окном

Для **middle** и **senior**:

Я рекомендую вам решить 30-50 задач из списка [LeetCode 75](#). Я также занимался по роадмапу от NeetCode ([NeetCode150](#)).

Если вы метите в конкретную компанию, то я рекомендую готовиться по списку частых вопросов именно в эту компанию. Бесплатный список можно найти в [этом репозитории](#):

 yandex_1year.csv	Chore: Add headers to all the files (#54)	5 months ago
 yandex_2year.csv	Chore: Add headers to all the files (#54)	5 months ago
 yandex_6months.csv	Chore: Add headers to all the files (#54)	5 months ago
 yandex_alltime.csv	Chore: Add headers to all the files (#54)	5 months ago

Заключение

Большое спасибо за прочтение! Если вам хочется посмотреть, как я прохожу алгоритмические собеседования в крупные российские компании на уровни middle+ и senior, можете поддержать меня на [Boosty](#).

Подписывайтесь на [YouTube канал](#), [телеграм канал](#) и [телеграм сообщество](#).

Желаю успехов в прохождении собеседований,

Артём Шумейко