

### ИСПОЛЬЗОВАНИЕ РЕСУРСА LEETCODE ПРИ ПОДГОТОВКЕ ИТ-СПЕЦИАЛИСТОВ

Изучение алгоритмов и структур данных является важнейшей и неотъемлемой частью подготовки ИТ-специалистов. В большинстве университетов мира для студентов ИТ-специальностей читаются отдельные курсы по алгоритмам и структурам данных или же данная проблематика рассматривается в параллельных курсах. И хотя порой в среде программистов или Интернет-пространстве возникают дискуссии о целесообразности изучения алгоритмов и структур данных, такая необходимость очевидна.

В крупных ИТ-компаниях, таких как Google, Amazon, IBM, Oracle, Яндекс, Microsoft и других ИТ-гигантах, алгоритмическое собеседование – обязательный этап отбора разработчиков. На нём проверяют умение быстро отразить идею в коде. Но знание алгоритмов требуют не только ИТ-гиганты – для многих компаний это базовый навык хорошего инженера. Зачастую задача успешного прохождения собеседований является основным стимулом к изучению алгоритмов и структур данных для программиста.

Существует множество ресурсов для изучения данной темы. Это и топовые книги по алгоритмам и структурам данных, и различные онлайн-курсы, например, курс Algorithms от Coursera, и специализированные веб-сайты, и Github библиотеки, и готовые подборки ресурсов для изучения алгоритмов и структур данных, публикуемые экспертами [1].

Одним из характерных ресурсов для практики и совершенствования навыков по программированию является LeetCode [2, 3]. Это сервис с алгоритмическими задачами, которые помогут подготовиться к собеседованию. Задачи охватывают все аспекты разработки, включая базы данных, алгоритмы, теорию графов, структуры данных и динамическое программирование. Кроме задач, доступны обучающие планы, чтобы освоить тему с нуля. Для специалистов доступны инструменты, которые помогут тестировать, отлаживать и даже писать собственные проекты онлайн. Ресурс LeetCode имеет большие возможности для обучения, и разработчик самостоятельно выбирает стратегию обучения. Остановимся на положительных аспектах LeetCode по мнению разработчиков:

- сервис помогает найти путь решения проблемы, а не просто подобрать подходящий алгоритм;
- повышается скорость закрытия простых задач [2];
- количество ошибок становится гораздо меньше, ведь перед тем, как отправить готовый ответ, нужно по несколько раз проверить правильность решения;

– легче учить другие языки сразу на практике, параллельно погружаясь в теорию в виде материалов на данном сайте;

– развитие самодисциплины [4].

*Задачи использования ресурса LeetCode в обучении студентов IT-специальностей.* Как уже отмечалось ранее, ресурс в основном используется для подготовки программистов к техническим собеседованиям. Вторым аспектом его использования является самостоятельное изучение алгоритмов и структур данных и, как следствие, повышение уровня решения алгоритмических задач конкретного программиста.

Можно ли использовать ресурс в процессе обучения студентов IT-специальностей? Безусловно. Отметим задачи, которые решаются в процессе обучения: более детальное изучение алгоритмов и структур данных; получение практики решения прикладных алгоритмических задач; получение навыка тестирования программ; получение практики написания «чистого кода»; умение оценивать временную и емкостную сложность алгоритмических задач; повышение уровня английского языка.

Раскроем и обоснуем указанные пункты. Практически для всех студентов IT-специальностей предусмотрены курсы по изучению алгоритмов и структур данных. Например, для специальности 6-05-0612-01 Программная инженерия в БрГТУ курс так и называется «Алгоритмы и структуры данных». Охватить и детально рассмотреть все аспекты структур данных и алгоритмов в годичном курсе просто невозможно, курс построен скорее «вширь», что вполне обосновано. Ресурс как раз и позволяет начать изучение «вглубь», раскрывая в конкретной задаче нюансы и возможности применения структур данных и алгоритмов. *Тестирование* – это важная часть написания программы (программного продукта). Обычно студент, написав программу, проверяет ее на 3-4 тестовых примерах, выбор которых на данный момент соответствует уровню его знаний и понимания решаемой задачи, порой не учитывая все граничные условия, исключения или варианты входных данных. Ресурс же «прогоняет» написанную программу на десятках вариациях входных данных, и может оказаться, что не прошел, к примеру, единственный тест из 40. А это и заставляет студента рассматривать проблему «вглубь», искать, какие нюансы он не учел.

Несмотря на очевидные плюсы использования ресурса в процессе изучения алгоритмических курсов, следует сделать два замечания.

– если дать студенту какую-либо задачу с ресурса, то студент 2-го курса скорее всего самостоятельно ее не решит, в крайнем случае решит ее «в лоб»;

– преподаватель должен либо сам уметь решать задачи с ресурса, а это порой олимпийский уровень программирования, либо составить базу разобранных и решенных задач с ресурса, используя раздел Discuss либо описание задач с других Интернет-ресурсов.

Предположим, что вторая проблема решена и осталась только задача обучения студентов. И тут возможны несколько вариантов или комбинации этих вариантов.

1. Преподаватель описывает алгоритм. Предпочтительным является еще и диалоговое обсуждение алгоритма. Студент должен понять алгоритм, написать оптимальную программу и успешно протестировать ее.

2. Приводится оптимальный код программы, принятый ресурсом, и студенту предлагается описать по пунктам алгоритм решения задачи. Студент обучается читать чужой код, что по началу может быть и нелегко и что является необходимым навыком каждого программиста. И, конечно, при данном варианте студент изучает алгоритм и учится писать «чистый код».

3. Студенту дается условие (порядковый номер) задачи, предлагается найти оптимальный алгоритм решения задачи, реализовать его и успешно протестировать. Поскольку, как уже отмечалось, самостоятельное решение алгоритмических задач для студентов может быть сложным, студент может зайти в раздел Discuss и посмотреть предлагаемые решения с их описанием. Описание ведется на английском языке, и его приводит конкретный программист. Также может приводиться описание алгоритма и кода программы, где строчка кода либо пропущена, либо указана неверно.

Вышеуказанные подходы использования ресурса LeetCode активно применяются рядом преподавателей кафедры интеллектуальных информационных технологий БрГТУ в рамках изучения курса «Алгоритмы и структуры данных» специальности 6-05-0612-01 Программная инженерия. Выбор конкретных вариантов использования LeetCode для обучения студентов диктуется целями, которые преподаватель ставит в лабораторной работе, его креативностью и творческим подходом. В любом случае, практика применения LeetCode продемонстрировала его несомненную эффективность: это и умение решать все более сложные алгоритмические задачи, и увеличение скорости написания стандартных заданий, и уменьшение количества ошибок, и практика «чистого» кода.

#### Список использованных источников

1. Зачем разработчику знать алгоритмы и структуры данных? [Электронный ресурс] / proglib. io. – Режим доступа: <https://proglib.io/p/zachem-razrabotchiku-znat-algoritmy-i-struktury-dannyh-2022-06-08>. – Дата доступа: 23.10.2023.

2. Тренажер программирования LeetCode – что это и как его использовать [Электронный ресурс] / blog.tutortop. – Режим доступа: <https://blog.tutortop.ru/trenazher-programmirovania-leetcode>. – Дата доступа: 23.10.2023.

3. Как правильно решать задачи на LeetCode: подробный гайд по тренажеру для программистов [Электронный ресурс] / skillbox.ru. – Режим доступа: <https://skillbox.ru/media/code/kak-pravilno-reshat-zadachi-na-leetcode-podrobnyy-gayd-po-trenazheru-dlya-programmistov>. – Дата доступа: 23.10.2023.

4. Как задачи на LeetCode прокачали меня как разработчика, или по-честному про алгоритмы [Электронный ресурс] / habr.com. – Режим доступа: <https://habr.com/ru/articles/747970>. – Дата доступа: 23.10.2023.