



---

# Содержание

<b>Предисловие</b> .....	9
<b>1. Разница курсов акций</b> .....	19
1.1. Алгоритмы .....	22
1.2. Время выполнения работы и сложность .....	27
1.3. Используем стек для разницы курсов акций .....	36
Примечания .....	44
Упражнения .....	45
<b>2. Исследуя лабиринты</b> .....	47
2.1. Графы .....	49
2.2. Представление графов .....	55
2.3. Обход графа в глубину .....	64
2.4. Поиск в ширину .....	76
Примечания .....	82
Упражнения .....	83
<b>3. Сжатие</b> .....	86
3.1. Сжатие .....	90
3.2. Деревья и очереди с приоритетом .....	94
3.3. Код Хаффмана .....	97
3.4. Сжатие Лемпеля — Зива — Велча .....	107
Примечания .....	122
Упражнения .....	123
<b>4. Секреты</b> .....	126
4.1. Попробуйте расшифровать .....	127
4.2. Шифрование одноразовым ключом, или одноразовый блокнот .....	133
4.3. AES-шифрование .....	138
4.4. Обмен ключами по методу Диффи — Хеллмана .....	148
4.5. Быстрое возведение в степень и возведение в степень по модулю .....	153
Примечания .....	159
Упражнения .....	161

<b>5. Делимся секретами.</b> . . . . .	162
5.1. Шифрование с открытым ключом . . . . .	163
5.2. Криптосистема RSA . . . . .	167
5.3. Хеширование сообщения . . . . .	179
5.4. Анонимизация интернет-трафика . . . . .	180
Примечания. . . . .	188
Упражнения. . . . .	189
<b>6. Все по порядку.</b> . . . . .	190
6.1. Топологическая сортировка. . . . .	192
6.2. Взвешенные графы. . . . .	198
6.3. Критические пути. . . . .	200
Примечания. . . . .	209
Упражнения. . . . .	210
<b>7. Строки, абзацы, пути</b> . . . . .	211
7.1. Кратчайшие пути. . . . .	215
7.2. Алгоритм Дейкстры . . . . .	218
Примечания. . . . .	227
Упражнения. . . . .	229
<b>8. Маршрутизация, арбитраж</b> . . . . .	230
8.1. Маршрутизация Интернета. . . . .	235
8.2. Алгоритм Беллмана — Форда (— Мура) . . . . .	241
8.3. Отрицательные веса и циклы . . . . .	249
8.4. Арбитраж . . . . .	253
Примечания. . . . .	258
<b>9. Что важнее всего.</b> . . . . .	259
9.1. Суть пейдж-ранка . . . . .	260
9.2. Матрица гиперссылок . . . . .	262
9.3. Степенной метод . . . . .	265
9.4. Матрица «Гугла» . . . . .	270
Примечания. . . . .	275
<b>10. Прочность голосования.</b> . . . . .	276
10.1. Система голосования . . . . .	277
10.2. Метод Шульце . . . . .	281
10.3. Алгоритм Флойда — Уоршелла. . . . .	295
Примечания. . . . .	297

<b>11. Методы перебора, невесты и дихотомии</b> . . . . .	298
11.1. Последовательный поиск . . . . .	299
11.2. Соответствие, сравнение, записи, ключи. . . . .	301
11.3. Эффект Матфея и степенные законы . . . . .	304
11.4. Самоорганизующийся поиск. . . . .	311
11.5. Задача о разборчивой невесте . . . . .	316
11.6. Бинарный поиск . . . . .	320
11.7. Представление целых чисел на компьютере . . . . .	325
11.8. И снова к бинарному поиску . . . . .	331
11.9. Деревья сравнений . . . . .	333
Примечания. . . . .	339
<b>12. Сортировочный компот</b> . . . . .	342
12.1. Сортировка методом выбора. . . . .	343
12.2. Сортировка методом вставок . . . . .	348
12.3. Сортировка кучей. . . . .	354
12.4. Сортировка слиянием. . . . .	364
12.5. Быстрая сортировка. . . . .	377
12.6. Богатство выбора . . . . .	386
Примечания. . . . .	390
Упражнения . . . . .	390
<b>13. Гардероб, коллизия и слот</b> . . . . .	393
13.1. Соотнесение ключей и значений . . . . .	394
13.2. Хеширование . . . . .	399
13.3. Функции хеширования. . . . .	402
13.4. Представление чисел с плавающей запятой и хеширование . . . . .	411
13.5. Коллизии . . . . .	415
13.6. Цифровые отпечатки. . . . .	426
13.7. Фильтры Блума . . . . .	431
Примечания. . . . .	445
Упражнения . . . . .	447
<b>14. Биты и деревья</b> . . . . .	449
14.1. Предсказание как проблема коммуникации. . . . .	450
14.2. Информация и энтропия . . . . .	453
14.3. Классификация . . . . .	460
14.4. Деревья решений. . . . .	463

14.5. Выбор атрибутов .....	468
14.6. Алгоритм ID3 .....	474
14.7. Основные принципы работы .....	482
14.8. Бритва Оккама .....	490
14.9. Затраты, проблемы, улучшения .....	492
Примечания .....	496
Упражнения .....	498
<b>15. Немного построчим .....</b>	<b>500</b>
15.1. Сравнение строк перебором .....	504
15.2. Алгоритм Кнута — Морриса — Пратта .....	507
15.3. Алгоритм Бойера — Мура — Хорспула .....	520
Примечания .....	529
Упражнения .....	530
<b>16. Предоставим дело случаю .....</b>	<b>532</b>
16.1. Случайные числа .....	535
16.2. Случайная выборка .....	545
16.3. Борьба за власть .....	553
16.4. В поисках простых чисел .....	566
Примечания .....	577
Упражнения .....	579
<b>Библиография .....</b>	<b>581</b>
<b>Предметный указатель .....</b>	<b>595</b>

---

# Предисловие

Подобно многим из моего поколения, я был воспитан на изречении: «Сатана всегда найдет недостойное дело для праздных рук». Будучи глубоко добродетельным ребенком, я верил всему, что мне говорили, и поэтому приобрел чувство моральной ответственности, вплоть до настоящего времени заставлявшее меня прилежно трудиться. Но хотя совесть и контролировала мои действия, взгляды мои претерпели серьезные изменения. Я утверждаю, что в мире сделано чересчур много работы, что вера в то, что работа — это добродетель, нанесла огромный вред и что в современных индустриальных странах следует проповедовать идеи, весьма далекие от тех, что издавна проповедовались.

---

Бертран Рассел, «Похвала праздности»

Моя книга посвящена алгоритмам; это такая вещь, которая показывает нам, что делать, чтобы ничего не делать. Можно сказать, это работа, которая помогает избежать еще большей работы. Из-за обилия изобретений, сделанных человеком, мы постоянно заполняем мозг большим количеством разнообразной информации. Благодаря же алгоритмам мы можем использовать свой мозг для размышлений.

Облегчение человеческого труда — благородная цель. Мы давно свыклись с мыслью, что машины призваны облегчить человеческий труд и сократить тяжкую, изнурительную работу, которая на протяжении многих веков считалась частью повседневного человеческого уклада. Мысль прекрасна, и нам стоит относить ее не только к физическому труду, но и к умственной работе. Однообразию и рутине напрочь убивают творческое мышление, а нам бы очень хотелось сохранить его. Вот здесь и приходят на помощь алгоритмы.

Современные цифровые технологии способны совершать операции, которые сам человек выполняет не задумываясь, ведь они

воплощают самую суть человеческой природы. Машины распознают и воспроизводят речь, переводят тексты, сортируют и приводят в порядок документы, предсказывают погоду, с поразительной точностью вылавливают закономерности в куче информации, заставляют работать другие машины, производят математические расчеты, становятся оппонентом в играх и помогают в изобретении новых механизмов — все благодаря алгоритмам. Работа алгоритмов облегчает нам работу и освобождает время на получение знаний в других областях, мы даже успеваем придумывать более сложные алгоритмы, которые еще сильнее уменьшают ежедневную рутину.

Алгоритмы появились задолго до компьютеров, еще в древние времена, и их применяют не только в области информационных технологий. Сложно найти сферу деятельности, которая обошлась бы без алгоритмов. Рано или поздно все люди сталкиваются с алгоритмами: однажды к ним приходит осознание, что алгоритмы — неотъемлемая часть их профессии, которая порой даже не связана с компьютерами, и тогда люди начинают интересоваться алгоритмами, стремясь оптимизировать свою работу.

Просто поразительно, как много сил мы тратим даже на повседневные мелкие дела, а все потому, что не знаем верного подхода. Каждый раз, когда вы ловите себя на мысли, что занимаетесь нудной, монотонной работой, скорее всего вы и правда попусту тратите время. Автору не раз доводилось наблюдать работу в офисах, когда сотрудники выполняли целый ряд однообразных действий, которые можно сделать за долю секунды и избежать волокиты, если только знать, как правильно организовать работу: нужно не отказываться от использования компьютеров (многие люди находят по десятку отговорок, чтобы не учиться ими пользоваться), а передать машинам выполнение всех однообразных задач.

---

## Целевая аудитория

Книга написана для тех, кто только начал знакомиться с алгоритмами. Если вы изучаете информатику, то данное руководство поможет вам постигнуть ее азы, а затем перейти к более сложной

литературе. Алгоритмы — основа программирования, и подобное введение способно охватить лишь малую часть темы и дать лишь общее представление о предмете.

Книга также окажется полезной людям, чья профессиональная или учебная деятельность хоть и не связана с информатикой, однако не обходится без алгоритмов. Почти в каждой дисциплине есть свои алгоритмы, и без них никуда; так что данное руководство расскажет о них тем — а таких людей немало, — кто желает их понять и сделать если не центром, то частью своей работы или учебы.

Книга придется по душе и тем, кто захочет прибегнуть к алгоритмам в простых, незначительных ситуациях, чтобы облегчить себе работу и избежать рутины; ведь задачу, выполнение которой обычно занимает несколько часов, можно решить за считанные секунды — стоит только написать программный код на современном скриптовом языке. Порой это становится настоящим откровением для непосвященных людей, которые полагают, что знание алгоритмов — прерогатива некой интеллектуальной элиты, хотя на самом деле алгоритмы доступны всем.

В наши дни никто всерьез не станет спорить с фактом, что без знания основ математики и науки сложно угнаться за современным миром и понять происходящие в нем процессы; точно так же и с алгоритмами: в современном социуме продуктивный труд невозможен, если не имеешь ни малейшего понятия об алгоритмах. Они лежат в основе человеческих жизни и опыта.

---

## Что нужно знать

Не стоит думать, что алгоритмы могут понять одни только программисты. Они состоят из четких инструкций по выполнению задач, и понять их способен каждый человек. В то же время для плодотворного труда и извлечения максимума пользы читатель должен обладать определенным багажом знаний.

Вовсе не обязательно быть опытным математиком, но иметь представление об основных математических понятиях и терминах необходимо. Уровень математики в данной книге не превы-



шает уровня старшей школы. Совсем не обязательно знать высшую математику, но понимать логику математических доказательств нужно, потому что точно таким же образом — с помощью логики — мы будем разбирать алгоритмы. Мы не будем делать сложные выкладки, однако читатель должен понимать, как строятся математические доказательства в этой книге.

Вовсе не обязательно быть профессиональным программистом, но нужно обладать базовыми навыками в работе с компьютером, знать, как пишутся программы, и понимать структуру языков программирования. От читателя не требуется глубокое понимание предмета; вообще-то лучше всего, если он только-только принялся его изучать. Компьютерные системы и алгоритмы очень близки и взаимно дополняют друг друга.

Также нужна любознательность. Алгоритмы призваны решать проблемы и делать это максимально плодотворно. Каждый раз, когда вы спрашиваете себя: «Как лучше решить ту или иную проблему?» — вы на самом деле ищете алгоритм.

---

## Как подан материал

Цель данной книги — объяснить алгоритмы доступным читателю языком, не принижая при этом его интеллекта. Если вы читаете книгу и понимаете, что ничего не понимаете, если в вашу голову закрадываются сомнения, что, возможно, она написана для гениев или профессионалов, если вы смущены тем, что прочитанные слова не складываются в общую картину, значит, книга оскорбляет ваш интеллект. Мы постараемся избежать любых неясностей. Нам придется что-то упростить, поэтому некоторые понятия, приведенные здесь, будут раскрыты не в полной, а лишь в необходимой мере.

Допущения и упрощение сложных вещей вовсе не означают, что читатель окажется избавлен от вдумчивого чтения, ведь мы не сомневаемся в его умственных способностях и не желаем их принижать. Мы полагаем, что читатель искренне жаждет разобраться в алгоритмах, процесс познания требует сил и времени, поэтому чем больше вы вникаете, тем больше извлекаете для себя пользы.

Обратимся на минутку к области литературы. Бывают книги, которые затягивают вас с головой и от которых невозможно оторваться, вы читаете их и не замечаете, как пролетает время. И речь тут не о модных бестселлерах. К примеру, «Чума» Альбера Камю читается очень легко, хотя это и очень серьезное и глубокое произведение.

Бывают книги, которые требуют значительного умственного труда. Их сложно читать, однако те люди, которым удается их понять, испытывают огромное удовольствие и даже чувство собственной уникальности от того, что смогли уловить мысль автора. Например, далеко не все любят «Улисса» Джеймса Джойса, книги Томаса Пинчона или Дэвида Фостера Уоллеса. В то же время вы вряд ли найдете хоть одного человека, который бы пожалел, что прочел их.

Есть третья категория книг, которая находится где-то между первыми двумя. Возможно, «Радуга земного тяготения» слишком специфична для вас, но разве можно сказать что-то подобное о «Братьях Карамазовых» или «Анне Карениной»?

Книга, которую вы держите в руках, как раз из тех, которые находятся где-то между: это не мудреный талмуд, но в то же время ее чтение требует некоторой сообразительности. Автор станет вашим проводником в мир алгоритмов, но не средством передвижения; он протянет вам руку и укажет путь через все алгоритмические хитросплетения, но пройти этот путь вы должны все же своими ногами. Ваш интеллект не будет оскорблен. Напротив, мы полагаем, что вы — любознательная личность, которая желает узнать много новых интересных вещей и прекрасно понимает, что в учении, как в бою, требуются усилия и отдача; что вы тот, кто знает — успех не приходит сам по себе, но добывается с помощью труда.

---

## Псевдокод

Годы назад молодые люди могли начать карьеру программиста, прекрасно владея всего одним программным языком. Теперь все изменилось. Сейчас в ходу множество хороших языков; компью-

теры теперь способны выполнять куда больше операций, чем двадцать лет назад, и в зависимости от конкретной задачи выбирается наиболее подходящий язык. Языковые войны глупы и бесполезны. К тому же из-за многообразия чудес, на которые способны нынешние компьютеры, люди все активнее взаимодействуют с ними, развивая новые сферы и направления, что приводит к созданию новых программных языков и эволюции старых.

Автор честно сознается, что считает некоторые языки более удобными, чем остальные, однако считает несправедливым и недопустимым навязывать читателю свои предпочтения. Более того, мода на языки меняется довольно быстро, и вчерашний фаворит может завтра оказаться не у дел. Чтобы книга стала максимально полезной и универсальной, на ее страницах не приводятся примеры каких-либо конкретных языков программирования — алгоритмы здесь описаны с помощью псевдокода. Его понять куда проще, чем какой-либо действующий язык программирования. С помощью псевдокода проще объяснить материал: когда вы пишете на нем, он позволяет сосредоточиться на понимании алгоритмов, а не на синтаксических конструкциях, которые постоянно требуют внимания и осторожности.

С другой стороны, сложно работать с алгоритмами без написания программного кода, с помощью которого они реализуются. Применение псевдокода в данном руководстве не означает, что читателю нужно так же легко и условно относиться к программным кодам в целом; по возможности следует реализовывать приведенные в книге алгоритмы, выбирая язык программирования на свой вкус. Не стоит забывать, что полное понимание и удовольствие от предмета будет достигнуто только тогда, когда вам удастся создать программу, которая реализует нужный алгоритм так, как вам требуется.

---

## Как читать данную книгу

Лучше читать все по порядку, одно за другим, так как в первых главах даются разъяснения и концепции, употребляемые в даль-

нейшем. В самом начале вы познакомитесь с основными структурами данных, которые используются во всех алгоритмах и проявят себя в следующих главах. Дальше, когда базис знаний будет заложен, вы можете читать главы в любом порядке, выбирая то, что вас больше интересует.

Таким образом, следует начать знакомство с самой первой главы, чтобы понять структуру всех последующих: сперва идет описание проблемы, а затем — разбор алгоритма, который позволяет ее решить. Также первая глава познакомит вас с условностями псевдокода данной книги, основными терминами и первыми структурами данных: массивами и стеками.

Вторая глава освещает тему графов и способы работы с ними. Также она рассказывает о рекурсии, так что, если вам уже доводилось видеть графы, но вы не до конца уловили идею рекурсии, рекомендуем не пропускать эту главу. Во второй главе мы также познакомимся с новыми структурами данных, которые время от времени будут нам попадаться в алгоритмах далее. В третьей главе мы рассмотрим проблему сжатия и то, как работают два различных метода сжатия: глава послужит нам подспорьем в понимании важных структур данных, с которыми мы встретимся позже.

Четвертая и пятая главы расскажут о криптографии. Она не похожа на графы и сжатие, однако является важной частью алгоритмов, особенно в последние годы, когда персональные данные стали храниться в различных местах и устройствах, вызывая у некоторых субъектов желание до них добраться. Эти две главы, в принципе, можно читать отдельно от остальных, хотя некоторые значимые фрагменты, например поиск больших простых чисел, помещены и рассмотрены в главе 16.

Главы 6–10 посвящены проблемам, связанным с графами: порядок заданий, нахождение выхода из лабиринта, определение приоритета соединенных друг с другом частей (например, страниц на сайте) и использование графов в выборах. Нахождение пути в лабиринте применяется куда чаще, чем может показаться на первый взгляд: от набора газетных статей до интернет-маршрутизации и валютного арбитража; его разновидность воз-

никает в контексте выборов, поэтому главы 7, 8 и 10 можно рассматривать как единое целое.

Главы 11 и 12 затрагивают две основные проблемы программирования: поиск и сортировку. По этим темам можно писать отдельные книги. Мы покажем несколько базовых и широко применяемых алгоритмов. Разобравшись с сортировкой, мы получим возможность разобраться с дополнительным материалом, например с интернет-поиском (поиск среди предметов, которые последовательно проходят перед вашими глазами, с условием, что после выбора нельзя изменить свое решение) и безмасштабной сетью, которую внимательный исследователь может увидеть где угодно. В главе 13 описан еще один путь сортировки и выборки данных — с помощью хеширования, очень полезного, распространенного и элегантного способа.

Глава 14 описывает классификацию алгоритмов: алгоритм учится группировать данные, основываясь на заданных примерах, а затем мы используем его, чтобы систематизировать новые, незнакомые случаи. Здесь мы сталкиваемся с машинным обучением, областью, значимость которой стремительно возрастает по мере того, как компьютеры становятся мощнее и мощнее. Глава также охватывает основные принципы теории информации, еще одной прекрасной области, тесно связанной с алгоритмами. Четырнадцатая глава отличается от остальных тем, что показывает, как алгоритм работает с более мелкими алгоритмами, выполняющими долю его работы; точно так же компьютерные программы состоят из небольших частей, каждая из которых имеет свое назначение. Глава так же показывает, каким образом структуры данных, описанные в данной книге, играют ключевую роль в выполнении алгоритма классификации. Эта глава особо заинтересует читателей, желающих узнать об устройстве частей высокоуровневых алгоритмов. Это станет большим шагом в понимании того, как алгоритмы превращаются в программы.

Глава 15 рассказывает о последовательности символов, именуемых также строками символов, и том, как в них что-то найти. Тут мы встретим операцию, к которой мы прибегаем каждый

раз, когда ищем что-то внутри текста и хотим добиться плодотворного результата. К счастью, существует несколько быстрых и изящных способов сделать это. Более того, последовательности символов могут представлять множество других вещей, так что сравнение строк применяется в самых разных областях, например в биологии.

В последней, шестнадцатой главе мы затронем алгоритмы, которые работают со случайностями. Существует поразительно большое количество таких рандомизированных алгоритмов, поэтому мы рассмотрим лишь небольшую их группу. Ко всему прочему они дадут нам ответы на поставленные в предыдущих главах вопросы, например поиск больших простых чисел в криптографии. Или опять же выборы: как подсчитать результаты.

---

## **Использование в обучении**

Материал данной книги рассчитан на семестровый курс, знакомящий учащихся с алгоритмами и объясняющий их основные принципы, не углубляясь в технические подробности. Студенты, изучающие самые разные дисциплины: бизнес и экономику, социальные и прикладные науки или же классические науки вроде математики и статистики, — могут использовать это руководство на начальной ступени обучения, как дополнение к практическому курсу программирования, чтобы сразу на практике оценить всю пользу алгоритмов. Те же, кто изучает информатику, могут использовать книгу для неформального знакомства с предметом: она позволит им оценить всю глубину и красоту алгоритмов, техническая сторона которых более подробно расписана в специализированных учебниках.

---

## **Заключительные вступительные слова**

Если взглядеться в слово «алгоритм», то можно увидеть в нем фразу «ритм боли», так как «алгос» на греческом значит «боль».

Но не стоит пугаться, слово «алгоритм» придумали не греки, а персидский математик, астроном и географ Аль-Хорезми (около 780–850 гг). Очень надеюсь, что данное руководство принесет вам радость познания и ни капли боли. Давайте же познакомимся с алгоритмами.

---

# 1 Разница курсов акций

Представьте, что курс акций меняется каждый день. Таким образом, у вас есть ряды чисел, каждое из которых представляет собой финальную стоимость конкретной акции в конкретный день. Дни идут в хронологическом порядке. Дни, когда фондовый рынок закрыт, не считаются.

Разница стоимостей акций в конкретный день — это число следующих друг за другом дней, от выбранного нами и в обратном направлении, до того дня, в который стоимость была меньше или равна стоимости в выбранный нами день. Таким образом, задача разницы курсов акций представляет следующее условие: даны ряды ежедневных курсов акции, и нам требуется найти разницу стоимостей акций для каждого дня в ряду. Для примера обратимся к рисунку 1.1. Первый день считается за 0. На шестой день нашего примера разница составляет пять дней, на пятый день — четыре дня, а на четвертый день — один день.

В реальной жизни в рядах могут находиться тысячи дней, и нам нужно будет просчитать разницу для множества различных рядов, в каждом из которых описаны изменения для разных курсов акций. Мы хотим поручить эту задачу компьютеру.

Многие задачи, решение которых мы поручаем компьютеру, имеют, как правило, не один способ решения, а несколько. Какие-то решения лучше, какие-то хуже. Впрочем, в данном случае «лучше» — весьма условный термин, который по сути ничего не значит, ведь, когда мы говорим «лучше», мы подразумеваем какой-то контекст: лучше в чем-то конкретном. Речь может быть о скорости, памяти или чем-то еще, что связано с такими ресурсами, как время и пространство. Мы еще коснемся данной темы, однако сейчас главное усвоить сказанное выше, потому что решение задачи может оказаться простым и в то же