ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

Компьютеры преобразуют наше общество не в меньшей степени, чем достижения физики и химии в два предшествующих столетия. Действительно, вряд ли остался в нашей жизни уголок, который не затронут – а чаще изменен до неузнаваемости – цифровыми технологиями. И коль скоро компьютеры так важны для современного общества, не странно ли, что люди так мало знают об основополагающих идеях, благодаря которым все это стало возможно? Изучение этих идей составляет основу информатики, а новая книга Маккормика – одна из немногих попыток донести их до широкой аудитории.

Одна из причин недооценки информатики как научной дисциплины заключается в том, что ее редко преподают в средней школе. Если начальный курс по таким предметам, как физика и химия, принято считать обязательным, то информатику как отдельную дисциплину обычно изучают только в колледжах или университетах. А если уж в школе есть предмет «информатика» или «информационные и телекоммуникационные технологии», то сводится он к приобретению навыков работы с некоторыми программными пакетами. Неудивительно, что ученикам это кажется скучным, а их естественная склонность к использованию компьютеров для развлечений и общения умеряется впечатлением, будто созданию таких технологий недостает интеллектуальной глубины. Есть мнение, что именно такое умонастроение стало причиной снижения количества студентов, изучающих информатику в университетах, на 50 % за последние 10 лет. Учитывая критическое значение цифровых технологий для современного общества, крайне важно возродить интерес населения к обаянию информатики.

В 2008 году мне была оказана честь открывать 180 серию рождественских лекций Королевского института, начало которым положил Майкл Фарадей в 1826 году. Лекции 2008 года впервые были посвящены теме информатики. Готовясь к ним, я много размышлял о том, как объяснить смысл информатики непрофессиональной аудитории, и обнаружил, что существует очень мало ресурсов и почти никаких

научно-популярных книг на эту тему. Поэтому новая книга Маккормика пришлась очень кстати.

Маккормик проделал великолепную работу и сумел изложить сложные идеи информатики в виде, доступном для широкой публики. Многие из этих идей необычайно красивы и элегантны, что само по себе делает их достойными внимания. Приведу всего один пример: бурный рост Интернет-торговли стал реальностью только потому, что появилась возможность безопасно посылать по сети конфиденциальную информацию (в частности, номера кредитных карт). В течение многих десятилетий задача о безопасном обмене данными по «открытым» каналам считалась неразрешимой. А когда решение было найдено, оно оказалось исключительно элегантным, и Маккормик объясняет его с помощью точных аналогий, не требующих предварительного знакомства с информатикой. Благодаря таким жемчужинам книга займет достойное место на полке с научно-популярной литературой, и я ее всячески рекомендую.

Крис Бишоп
заслуженный научный работник, *Microsoft Research, Кэмбридж*
вице-президент, *Королевский институт Великобритании*
профессор информатики, *Эдинбургский университет*

ГЛАВА 1. Введение: необычные идеи, каждодневно используемые в компьютерах

Природа дарования, коим я обладаю, очень-очень проста. Ум мой от рождения предрасположен к фантазии, причудливо выражающейся в образах, фигурах, формах, предметах, понятиях, представлениях, порывах и отступлениях.

— Вильям Шекспир «*Бесплодные усилия любви*»

Как появились на свет великие идеи информатики? Вот несколько примеров.

- В 1930-х годах, еще до создания первого цифрового компьютера, гениальный британский ученый открывает научную дисциплину информатики, а затем доказывает, что существуют задачи, которые не сможет решить ни один будущий компьютер, каким бы он ни был быстрым и мощным и как бы хитроумно ни был спроектирован.
- В 1948 году сотрудник телефонной компании публикует статью, которая заложила основы теории информации. Эта работа доказала, что компьютеры могут передавать сообщения без искажения, даже если большая часть данных изменена в результате воздействия помех.
- В 1956 году группа ученых приезжает на конференцию в Дартмуте с четко осознанной дерзкой идеей основать новую науку – искусственный интеллект. После многочисленных ярких достижений и не менее многочисленных глубочайших разочарований мы все еще ждем появления компьютерной программы, обладающей настоящим интеллектом.

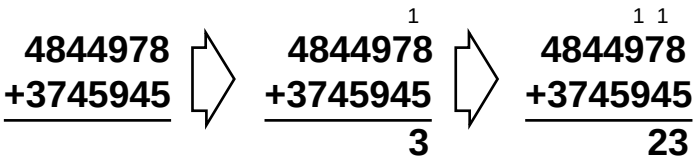
- В 1969 году исследователь из компании IBM открывает новый элегантный способ организовать информацию в базе данных. Сегодня эта технология применяется для хранения и извлечения информации, участвующей в большинстве сделок, совершаемых в онлайновом режиме.
- В 1974 году исследователи из финансируемой Британским правительством лаборатории по секретным коммуникациям открывают способ безопасного обмена данными между двумя компьютерами в ситуации, когда третий компьютер может наблюдать за всеми передаваемыми данными. Эти исследователи были связаны обязательством хранить государственную тайну, но по счастью три американских профессора независимо открыли и развили эту идею, которая ныне лежит в основе всех безопасных коммуникаций в Интернете.
- В 1996 году два студента Стэнфордского университета, работающих над диссертацией, решают объединить усилия и построить поисковую систему для веб. Спустя несколько лет они создали компанию Google – первый цифровой гигант эпохи Интернета.

Сегодня, в 21 веке мы наслаждаемся плодами поражающего воображение развития технологий, но никаких вычислительных устройств – будь то кластер из самых мощных доступных сегодня машин или модный гаджет последней модели – не существовало бы без основополагающих идей информатики, возникших в 20 столетии. Подумайте: делали ли вы сегодня что-то поразительное? Ответ, конечно, зависит от точки зрения. Быть может, вы перелопатили миллиарды документов в поисках двух-трех, которые содержат нужные сведения? Или сохранили либо передали по сети миллионы единиц информации без единой ошибки, несмотря на электромагнитные помехи, которым подвержены все электронные устройства? Или благополучно купили что-то через Интернет, хотя вместе с вами к тому же серверу обращались тысячи других пользователей? Или безопасно передали по проводам конфиденциальные данные (к примеру, номер своей кредитной карты), хотя вас могли подслушать десятки других компьютеров. А, быть может, вы воспользовались магией сжатия, позволяющий уменьшить размер многомегабайтной фотографии до величины, позволяющей передать ее по электронной почте? Или, даже не подозревая об этом, прибегли к скрытому в наладонном устройстве искусственному интеллекту, который исправляет опечатки, когда вы вводите текст с помощью его крохотной клавиатуры? Каждое из этих

деяний достойно изумления, и все они основаны на замечательных открытиях, о которых было сказано выше. Стало быть, любой пользователь компьютера каждый день по много раз применяет эти гениальные идеи, даже не осознавая этого! Цель этой книги – объяснить величайшие идеи информатики максимально широкой аудитории. Никакого предварительного знакомства с информатикой не предполагается.

Алгоритмы – чародейство услужливого джинна

До сих пор я говорил о великих «идеях» информатики, но сами ученые употребляют слово «алгоритм». Так в чем же разница между идеей и алгоритмом? Что вообще такое алгоритм? Простейший ответ звучит так: алгоритм – это рецепт, в котором описывается точная последовательность шагов, приводящая к решению задачи. Один такой алгоритм всем нам известен со школьной скамьи: сложение двух больших чисел. Пример его применения показан выше. Описание последовательности шагов этого алгоритма начинается так: «Сначала сложить две последние цифры, записать последнюю цифру результата и перенести избыток в следующий столбец слева; затем сложить цифры в следующем столбце и прибавить к ним перенесенный из предыдущего столбца избыток...» – и так далее.



Первые два шага алгоритма сложения двух чисел

Обратите внимание, что шаги этого алгоритма почти механические. И это одна из важнейших особенностей любого алгоритма: каждый шаг должен быть описан абсолютно точно, его выполнение не должно требовать от человека догадки или озарения. И тогда эти механические шаги можно будет оформить в виде программы для компьютера. Еще одна важная особенность алгоритма заключается в том, что он должен работать всегда, какие бы данные ни были поданы на вход. Алгоритм сложения, который мы учили в школе, обладает этим свойством: каковы бы ни были два складываемых числа, этот алго-

ритм рано или поздно даст правильный ответ. Так, его, безусловно, можно применить для сложения двух тысячезначных чисел, хотя на это и потребуется довольно много времени.

Возможно, у вас возник вопрос в связи с определением алгоритма как точного, механического рецепта. А настолько точным должен быть рецепт? Какие фундаментальные операции разрешены? Например, если речь идет об алгоритме сложения, то достаточно ли просто сказать «сложить две цифры» или нужно выписать полную таблицу сложения однозначных чисел? Такие детали могут показаться несущественными или даже излишне педантичными, но это представление очень далеко от истины: нахождение правильных ответов на такие вопросы как раз и составляет самую суть информатики, и здесь имеются связи с философией, физикой, неврологией и генетикой. Глубокие вопросы о том, что такое алгоритм на самом деле, в конечном счете сводятся к так называемому *тезису Чёрча-Тьюринга*. Мы вернемся к этой теме в главе 10, где будем обсуждать теоретические пределы вычислений и некоторые аспекты тезиса Чёрча-Тьюринга. А пока неформального представления об алгоритме как об очень точном рецепте нам будет вполне достаточно.

Итак, мы знаем, что такое алгоритм, но какое отношение это имеет к компьютерам? Суть дела в том, что для программирования компьютеров нужно формулировать инструкции очень точно. Прежде чем поручить компьютеру решение какой-то задачи, мы должны разработать соответствующий алгоритм. В других научных дисциплинах, например в математике и физике, важные результаты зачастую можно выразить одной формулой (из широко известных примеров можно назвать теорему Пифагора, $a^2 + b^2 = c^2$, или формулу Эйнштейна $E = mc^2$). Напротив, в информатике для формулирования важной идеи нужно описать, *как* решить задачу – разумеется, с применением какого-то алгоритма. Поэтому главная цель этой книги заключается в том, чтобы объяснить, что превращает ваш компьютер в личного джинна: великие алгоритмы, лежащие в основе его работы.

Какой алгоритм считать великим?

Это подводит нас к непростому вопросу: какие алгоритмы действительно «великие»? Перечень потенциальных кандидатов обширен, поэтому чтобы уложиться в объем книги, я применил несколько критериев отбора. Первый и самый главный критерий: алгоритм должен

ежедневно использоваться в обычном компьютере. Второй важный критерий: алгоритм должен решать конкретную реально возникающую задачу, например, сжатие файла или его передачу по зашумленному каналу. Для читателей, немного знакомых с информатикой, во врезке ниже описаны некоторые следствия, вытекающие из этих двух критериев.

Третий критерий заключается в том, что алгоритмы должны относиться преимущественно к *теоретической* информатике. Это исключает все связанное с компьютерным оборудованием: процессорами, мониторами, сетями и т. д. Кроме того, это означает, что устройству инфраструктуры, например Интернета, мы уделяем меньше внимания. Почему я решил сосредоточиться на теоретических основах информатики? Отчасти потому что существует перекос в общественном восприятии информатики: широко распространено мнение, будто информатика сводится в основном к программированию (то есть «софту») и конструированию оборудования («железа»). На самом же деле, многие из самых красивых идей информатики совершенно абстрактны и не попадают ни в одну из этих категорий. Ставя во главу угла теоретические идеи, я льщу себя надеждой, что больше людей придут к пониманию природы информатики как интеллектуальной дисциплины.

Вы, наверное, заметили, что мои критерии призваны отсеять потенциально великие алгоритмы, но я не ответил на гораздо более трудный вопрос: чем вообще определяется величие? Тут я решил положиться на собственную интуицию. В основе любого рассматриваемого в этой книге алгоритма лежит некий остроумный трюк, благодаря которому все и работает. Именно желание воскликнуть «ага!», когда трюк становится понятен, и доставляло мне радость, когда я объяснял алгоритм, и, надеюсь, доставит вам не меньшую радость, когда вы будет это объяснение читать. Поскольку я часто буду употреблять слово «трюк», то хочу сразу сказать, что не имею в виду ничего постыдного или мошеннического – такого, что вытворяют шулеры за карточным столом. Скорее, речь идет о профессиональных приемах или трюках фокусника: хитроумной технике, позволяющей достичь цели, когда другие способы затруднительны или даже невозможны.

Первый критерий – повседневное использование обычным пользователем компьютера – исключает алгоритмы, применяемые в основном профессионалами, например, компиляторы и способы верификации программ. Второй критерий – применение к конкретной задаче – исключает многие великие алгоритмы, изучаемые на млад-

ших курсах факультетов информатики. К ним относятся алгоритмы сортировки, например быстрая сортировка, алгоритмы на графах, в том числе алгоритм Дейкстры для поиска кратчайшего пути, и различные структуры данных, например хэш-таблицы. Нет сомнения, что это великие алгоритмы и что они отвечают первому критерию, поскольку активно используются в большинстве программ, запускаемых обычными пользователями. Но это алгоритмы общие: их можно применить к самым разным задачам. А в этой книге я решил сосредоточиться на алгоритмах решения конкретных задач, так как считаю, что это будет интереснее неискушенным пользователям.

Дополнительные сведения о принципах отбора алгоритмов для этой книги. Вообще говоря, не предполагается, что читатели знакомы с информатикой. Но если у вас есть познания в этой области, то из этой врезки станет ясно, почему многие из ваших любимых алгоритмов в книгу не попали.

Итак, я положился на собственную интуицию при выборе того, что мне кажется самыми остроумными трюками в мире информатики. Английский математик Г. Х. Харди в своей книге «Апология математика», пытаясь объяснить публике, почему математики занимаются своей наукой, выразил эту мысль такими словами: «Красота служит первым критерием: в мире нет места безобразной математике». Тот же критерий красоты применим и к теоретическим основам информатики. Таким образом, последний критерий отбора алгоритмов для этой книги – критерий красоты Харди. Надеюсь, что мне хотя бы отчасти удалось передать то ощущение красоты, который я испытывал, объясняя вошедшие в книгу алгоритмы.

Но перейдем к отобранным мной алгоритмам. Всепроникающие поисковые системы – пожалуй, самый наглядный пример алгоритмической технологии, которая оказывает влияние на всех пользователей компьютеров, поэтому неудивительно, что я включил несколько базовых алгоритмов поиска в Интернете. В главе 2 описывается, как поисковые системы используют *индексирование*, чтобы находить документы, отвечающие запросу, а в главе 3 объясняется алгоритм *PageRank*, который применялся в первой версии Google для того, чтобы поместить самые релевантные документы в начало списка результатов поиска.

Даже не задумываясь надолго над этим вопросом, большинство из нас *нутром чуют*, что своими поистине чудесными результатами поисковые системы обязаны каким-то глубоким научным идеям. Но есть и другие великие алгоритмы, которые часто вызываются, хотя пользователь об этом даже не подозревает. Примером такого алгоритма является криптография с открытым ключом, которой посвящена

глава 4. Всякий раз, заходя на защищенный сайт (адрес которого начинается с `https`, а не с `http`), вы пользуетесь так называемым алгоритмом *обмена ключами*, чтобы организовать защищенный сеанс. А это как раз часть криптографии с открытым ключом. В главе 4 объясняется, как устроен этот обмен ключами.

В главе 5 рассматриваются коды, исправляющие ошибки, – еще один класс алгоритмов, которыми мы пользуемся, даже не задумываясь об этом. На самом деле, если попытаться назвать какую-то одну великую идею, которая используется чаще всего, то это, пожалуй, будут коды, исправляющие ошибки. Благодаря им компьютер может обнаруживать *и исправлять* ошибки при хранении или передаче данных, не прибегая к резервному копированию или повторной передаче. Эти коды присутствуют всюду: во всех жестких дисках, во многих механизмах передачи по сети, на CD и DVD и даже в некоторых устройствах памяти. Но они так хорошо справляются со своей работой, что мы о них и не подозреваем.

Глава 6 выбивается из общего ряда. В ней обсуждаются алгоритмы распознавания образов, которые вкрались в список великих идей информатики, хотя и нарушают первый критерий: повседневное применение обычными пользователями. Распознавание образов – это целый класс алгоритмов, с помощью которых компьютеры выделяют значимые признаки из информации, характеризующейся высокой степень изменчивости: рукописный текст, речь, лица людей. На самом деле, в первой декаде 21 века в повседневных компьютерных приложениях эти методы не использовались. Но в 2011 году, когда я пишу эти слова, важность распознавания образов стремительно возрастает: в мобильных устройствах с миниатюрными экранными клавиатурами необходимо автоматически исправлять опечатки, планшеты должны распознавать рукописный ввод, и все больше подобных устройств (особенно смартфонов) управляются голосом. На некоторых вебсайтах распознавание образов даже применяется, чтобы показывать пользователю интересную ему рекламу. Ко всему прочему, я питаю особую склонность к распознаванию образов, потому что сам работаю в этой области. В силу указанных причин в главе 6 рассматриваются три наиболее интересных и успешно применяемых алгоритма распознавания образов: классификаторы по ближайшим соседям, деревья решений и нейронные сети.

Алгоритмы сжатия – тема главы 5 – еще один кластер великих идей, благодаря которым компьютер превращается в услужливого джинна. Иногда пользователи применяют сжатие самостоятельно,

например, чтобы сэкономить место на диске или уменьшить размер фотографии перед отправкой ее по электронной почте. Но гораздо чаще сжатие используется подспудно: хотя мы этого не знаем, скачиваемые или закачиваемые данные сжимаются для экономии полосы пропускания, а центры обработки данных нередко сжимают пользовательские данные для снижения затрат. Те 5 ГБ, которые почтовый провайдер предоставляет вам бесплатно, на его диске, скорее всего, занимают куда меньше места!

В главе 8 мы рассмотрим кое-какие фундаментальные алгоритмы, лежащие в основе работы баз данных. Мы уделим особое внимание хитроумным методам обеспечения *непротиворечивости* данных, хранящихся в базе. Без них наша жизнь в онлайне (в том числе покупки в Интернет-магазинах и общение в социальных сетях типа Facebook) погрязла бы в нагромождении машинных ошибок. В этой главе я объясню, в чем состоит проблема непротиворечивости и как ученые решают ее, не жертвуя феноменальной эффективностью, которой мы ожидаем от онлайновых систем.

В главе 9 речь пойдет об одной из неоспоримых жемчужин теоретической информатики: цифровой подписи. На первый взгляд кажется, что «подписать» электронный документ цифровым способом невозможно. Ну, действительно – такая подпись по определению состоит из цифровой информации, которую легко может скопировать любой желающий подделать подпись. Разрешение этого парадокса стало одним из самых значительных достижений информатики.

В главе 10 мы повернем совсем в другую сторону: вместо того чтобы описывать уже существующий великий алгоритм, мы поговорим об алгоритме, который *был бы* великим, если бы существовал. Как это ни удивительно, мы обнаружим, что этот конкретный великий алгоритм невозможен. Это полагает некие абсолютные пределы умению компьютеров решать задачи, и мы вкратце обсудим следствия этого результата для философии и биологии.

В заключение мы сведем воедино некоторые общие черты великих алгоритмов и поразмышляем о возможном будущем. Существуют ли еще какие-нибудь великие алгоритмы или мы уже открыли все?

Сейчас удобный момент упомянуть об одной особенности стиля этой книги. В научной литературе принято указывать источники, но изобилие цитат нарушило бы ритм текста и придало бы ему излишнюю академичность. Поскольку понятность и удобочитаемость стояли для меня на первом месте, я не стал включать цитаты в основной текст. Однако же все источники скрупулезно перечислены – зачас-

тую даже с дополнительными замечаниями – в разделе «Источники и литература для дальнейшего чтения» в конце книги. Там интересующиеся читатели найдут ссылки на дополнительные материалы, из которых смогут узнать гораздо больше о великих алгоритмах информатики.

И раз уж зашла речь об особенностях, упомяну заодно о небольшой поэтической вольности в названии книги. Наши «Девять алгоритмов, которые изменили мир» без сомнения революционны, но действительно ли их девять? Это вопрос спорный, ответ на него зависит от того, что считать отдельным алгоритмом. Посмотрим, откуда взялось число «девять». Если исключить введение и послесловие, то в книге девять глав, и в каждой обсуждаются алгоритмы, оказавшие революционное влияние на различные типы вычислительных задач: криптографию, сжатие, распознавание образов и т. д. Таким образом, «девять алгоритмов» – это в действительности девять классов алгоритмов для решения девяти разных проблем.

А какое нам, собственно, дело до великих алгоритмов?

Хочется надеяться, что этот краткий обзор пленительных идей заронил в вас желание копнуть глубже и понять, как они все-таки работают. Но, возможно, вы все еще недоумеваете: какова же конечная цель? Позвольте мне сказать несколько слов об истинном назначении этой книги. Определенно, это не техническое руководство. Прочитав книгу, вы не станете специалистом по компьютерной безопасности, искусственному интеллекту или еще какой-то области информатики. Конечно, кое-какие практически полезные знания вы приобретете. Например, вы будете знать, как проверить атрибуты «безопасных» веб-сайтов и «подписанных» программных пакетов. Вы также сможете осознанно решать, какой алгоритм сжатия – с потерей или без потери информации – лучше подходит для конкретной задачи. И, возможно, понимая некоторые особенности индексирования и ранжирования страниц, вы станете более эффективно пользоваться поисковыми системами.

Но это все мелочи по сравнению с главной целью книги. Прочитав ее, вы *не* станете гораздо более опытным пользователем компьютера. Но вы сумеете по-настоящему оценить красоту идей, с которыми имеете дело каждый день, пользуясь своими компьютерными устройствами.

Почему это хорошо? Попробую прибегнуть к аналогии. Я, безусловно, не специалист по астрономии – больше того, я почти полный невежда в этой области и хотел бы знать больше. Но всякий раз как я гляжу на ночное небо, те слабые познания в астрономии, которые у меня все-таки есть, усиливают наслаждение от этого времяпрепровождения. Почему-то понимание того, что я вижу, вселяет ощущение изумления и благоговения. Я очень надеюсь, что, прочитав эту книгу, вы хотя бы иногда будете испытывать то же чувство изумления и благоговения при работе с компьютером. Вы научитесь по достоинству ценить самый вездесущий и непроницаемый черный ящик всех времен: свой персональный компьютер, своего услужливого джинна.