
Серия основана в 1999 году.

Библиотека
«Математическое просвещение»
Выпуск 29

С. Б. Гашков

СИСТЕМЫ СЧИСЛЕНИЯ И ИХ ПРИМЕНЕНИЕ

Издательство Московского центра
непрерывного математического образования
Москва • 2004

УДК 511.1
ББК 22.130
Г12

Аннотация

Различные системы счисления используются всегда, когда появляется потребность в числовых расчётах, начиная с вычислений младшеклассника, выполняемых карандашом на бумаге, кончая вычислениями, выполняемыми на суперкомпьютерах.

В книжке кратко изложены и занимательно описаны некоторые из наиболее популярных систем счисления, история их возникновения, а также их применения, как старые, так и новые, как забавные, так и серьёзные.

Бóльшая её часть доступна школьникам 7—8 классов, но и опытный читатель может найти в ней кое-что новое для себя.

Текст книжки написан на основе лекций, прочитанных автором в школе им. А. Н. Колмогорова при МГУ и на Малом мехмате МГУ.

Книжка рассчитана на широкий круг читателей, интересующихся математикой: школьников, учителей.

Издание осуществлено при поддержке
Московской городской Думы
и Департамента образования г. Москвы.

ISBN 5-94057-146-8

Сергей Борисович Гашков.

Системы счисления и их применение.

(Серия: «Библиотека „Математическое просвещение“»).
М.: МЦНМО, 2004. — 52 с.: ил.

Редакторы *Е. В. Корицкая*, *Ю. Г. Кудряшов*.
Художник *Т. И. Котова*.
Техн. редактор *М. Ю. Панов*. Корректор *Т. Л. Коробкова*.

Лицензия ИД № 01335 от 24/III 2000 года. Подписано к печати 25/II 2004 года. Формат бумаги $60\times88\ {}^{1}/_{16}$. Офсетная бумага № 1. Офсетная печать. Физ. печ. л. 3,25. Усл. печ. л. 3,18. Уч.-изд. л. 3,62. Тираж 5000 экз. Заказ 6709.

Книга соответствует гигиеническим требованиям к учебным изданиям для общего и начального профессионального образования (заключение государственной санитарно-эпидемиологической службы Российской Федерации № 77.99.02.953.Д. 002797.04.03 от 18/IV 2003 года).

Издательство Московского центра непрерывного математического образования.
119002, Москва, Г-2, Бол. Власьевский пер., 11. Тел. 241 72 85, 241 05 00.

Отпечатано с готовых диапозитивов
в ФГУП «Производственно-издательский комбинат ВИНИТИ».
140010, г. Люберцы Московской обл., Октябрьский пр-т, 403. Тел. 554 21 86.

§ 1. ДЕНЬГИ В КОНВЕРТАХ И ЗЁРНА НА ШАХМАТНОЙ ДОСКЕ

Представьте себе, дорогой читатель, что вы банкир, занимающийся отмыванием грязных денег, и завтра ждёте важного клиента, которому вы должны выдать круглую или не очень круглую, но заранее вам неизвестную сумму от 1 до 1 000 000 000 у. е. Чтобы не пачкать руки о грязные деньги, вы заранее дали указание своим кассирам заготовить некоторое количество конвертов с деньгами, на которых написаны содержащиеся в них суммы, и собираетесь просто отдать клиенту один или несколько конвертов, в которых и будет содержаться требуемая им сумма. Какое наименьшее количество конвертов необходимо иметь?

Конечно, можно просто заготовить конверты со всеми суммами от 1 до 1 000 000 000. Но где взять столько денег на конверты?

1. А какова будет в этом случае полная сумма во всех конвертах? Попробуйте оценить также массу бумаги, предполагая, что использованы не более чем сотенные купюры*).

Есть более рациональный подход к нашему делу. Надо положить в первый конверт 1 у. е., а в каждый следующий класть вдвое большую сумму, чем в предыдущий. Тогда, например, в 5-м конверте будет 16 у. е., в 10-м — 512 у. е., в 11-м — 1024 у. е., в 21-м — $1024^2 = 1\,048\,576$ у. е., в 31-м — $1024^3 = 1\,073\,741\,824$ у. е., но он нам, очевидно, уже не понадобится, а вот 30-й с $1\,073\,741\,824/2 = 536\,870\,912$ у. е. может и пригодиться. В общем случае сумма в $(n+1)$-м конверте будет равна произведению n двоек, это число принято обозначать 2^n и называть n-й степенью двойки. Условимся считать, что $2^0 = 1$. Проведённые выше вычисления основывались на следующих свойствах операции возведения в степень:

$$2^n 2^m = 2^{n+m}, \quad 2^n / 2^m = 2^{n-m}, \quad (2^n)^m = 2^{nm}.$$

Экспериментально легко проверить, что любое число можно представить единственным образом в виде суммы различных меньших степеней двойки, и поэтому наша задача почти решена. Например,

$$30\,000 = 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^8 + 2^5 + 2^4.$$

Но для реального применения нужен алгоритм построения такого разложения. Далее будут приведены несколько разных алгоритмов, но вначале мы рассмотрим самый простой. В сущности, это алгоритм выдачи сдачи клиенту, записанный некогда даже в инструкции для работников торговли, но очень редко ими выполняющийся. А он очень прост — сдачу надо выдавать, начиная с самых больших купюр. В нашем случае нужно найти конверт с наибольшей суммой денег, не превосходящей требуемую, т. е. наибольшую степень двойки, не превосходящую требуемого количества денег.

*) Двумя чертами слева выделены тексты задач для самостоятельного решения.

Если требуемая сумма равна этой степени, то алгоритм заканчивает работу. В противном случае опять выбирается конверт с наибольшей суммой денег, не превосходящей оставшуюся, и т. д. Алгоритм закончит работу, когда останется сумма, в точности равная степени двойки, и она будет выдана последним конвертом.

Ниже мы докажем, что, имея набор конвертов с суммами в 1 у. е., 2 у. е., 4 у. е., ..., 2^n у. е., любую сумму денег от 1 у. е. до $2^{n+1}-1$ у. е. можно выдать единственным способом. Также будет доказано, что, действуя по описанному алгоритму, мы всегда получим этот способ выдачи требуемой суммы.

Вначале рассмотрим пример работы алгоритма с числом 2^n-1. Ясно, что на первом шаге будет выбрано число 2^{n-1}, останется число $2^n-1-2^{n-1}=2^{n-1}-1$, потом будет выбрано число 2^{n-2}, и т. д., и в результате получится разложение

$$2^n-1=2^{n-1}+2^{n-2}+\ldots+2^2+2^1+2^0.$$

Но оно не показалось бы очевидным, если, не зная заранее ответа, пришлось бы вычислять сумму

$$1+2+4+8+\ldots+2^{n-2}+2^{n-1},$$

называемую суммой геометрической прогрессии со знаменателем 2. Ведь для этого пришлось бы выдумать какой-нибудь трюк наподобие следующего:

$$1+2+4+8+\ldots+2^{n-2}+2^{n-1}=2-1+2+4+8+\ldots+2^{n-2}+2^{n-1}=$$
$$=4-1+4+8+\ldots+2^{n-2}+2^{n-1}=8-1+8+16+\ldots+2^{n-2}+2^{n-1}=\ldots$$
$$\ldots=2^{n-2}-1+2^{n-2}+2^{n-1}=2^{n-1}-1+2^{n-1}=2^n-1.$$

2. Используя подобный трюк, вычислите произведение

$$(2+1)(2^2+1)\cdot\ldots\cdot(2^{2^n}+1).$$

Докажем теперь существование и единственность представления числа N в виде суммы меньших степеней двойки. Доказательство будем проводить индукцией по N.

Для $N=1$ утверждение очевидно.

Пусть оно верно для всех $N<N_0$. Пусть 2^n — максимальная степень двойки, не превосходящая N, т. е. $2^n\leqslant N_0<2^{n+1}$. Тогда по предположению индукции число $N_0-2^n\leqslant 2^n$ представимо в виде суммы степеней двойки, меньших $N_0-2^n<2^n$. Следовательно, число N_0 тоже представимо в виде суммы меньших степеней двойки (достаточно к представлению числа N_0-2^n добавить 2^n). Кроме того, так как $1+2+\ldots+2^{n-1}=2^n-1<2^n$, то не существует представления числа N_0, не использующего 2^n. Таким образом, доказана единственность такого представления.

Заметим, что для быстрого применения этого алгоритма удобно заранее вычислить все степени двойки, не превосходящие данного числа.

Заметим ещё, что, в отличие от первого варианта решения, полная сумма во всех конвертах менее чем в два раза превосходит верхнюю границу подлежащей выплате суммы.

Для краткой записи результата работы алгоритма над данным числом a можно вместо разложения

$$a=2^{n_1}+\ldots+2^{n_k},$$

которое и записать-то в общем виде без использования трёхэтажных обозначений затруднительно, использовать последовательность показателей степеней $(n_1, \ldots, n_k)$, или, что ещё удобнее (но не всегда короче), написать последовательность $(a_m, \ldots, a_1)$ чисел 0 и 1, в которой $a_i=1$, если число 2^{i-1} входит в указанное выше разложение, и $a_i=0$ в противном случае. Тогда это разложение можно будет переписать в виде

$$a=a_1+2a_2+4a_3+\ldots+2^{m-1}a_m.$$

Ясно, что приведённый выше алгоритм позволяет строить такое представление, причём оно определяется однозначно, если предполагать, что старший его разряд a_m ненулевой. Это представление и называется двоичной записью числа a.

Читатель увидит, что понятие двоичной записи очень похоже на понятие десятичной записи и в каком-то смысле даже проще.

Остался вопрос о минимальности найденной системы конвертов. В общем виде указанный выше приём предлагает для уплаты любой суммы от 1 до n использовать m конвертов с суммами 1, 2, 4, 8, ..., 2^{m-1}, где $2^{m-1}\leqslant n<2^m$. Меньшего количества конвертов может не хватить, потому что с помощью $k<m$ конвертов можно уплатить не более чем $2^k-1<2^{m-1}\leqslant n$ разных сумм, так как каждая сумма однозначно определяется ненулевым набором $(a_1, \ldots, a_k)$, в котором каждое число a_i равно 1, если i-й конверт входит в эту сумму, и равно 0 в противном случае, а всего наборов длины k из нулей и единиц можно составить ровно 2^k.

3. Докажите последнее утверждение.

4. Докажите, что если $n=2^m-1$, то минимальная система конвертов определяется однозначно, в противном случае — нет.

После упоминания десятичной системы сразу возникает идея на первый взгляд даже более простого решения задачи о конвертах. Надо просто заготовить конверты с суммами 1, 2, ..., 9, 10, 20, 30, ..., 90, 100, 200, 300, ..., 100 000 000, 200 000 000, ..., 900 000 000. Тогда для выплаты любой требуемой суммы не нужно искать её двоичную запись, так как для выплаты, например, 123 456 789 у. е. нужно просто взять конверт с суммой 9, конверт с суммой 80, конверт с суммой 700 и т. д. Это действительно проще, но исключительно потому, что мы привыкли пользоваться десятичной системой и все расчёты ведутся с её помощью. Если бы мы использовали в повседневной жизни только двоичную систему,

то этот способ был бы сложнее, так как приходилось бы переводить данную сумму из двоичной системы в десятичную*). Поэтому простота десятичного способа решения задачи скорее мнимая.

На самом деле указанный выше двоичный метод имеет преимущество перед десятичным (и любым другим). Оно заключается в меньшем числе используемых конвертов, что было показано выше. Хотя длина двоичной записи числа в три с лишним раза больше длины его десятичной записи, на каждую цифру десятичной записи приходится девять конвертов, т. е. число конвертов в двоичном методе почти в три раза меньше, чем в десятичном.

Идея, лежащая в основе изложенной задачи, видимо, очень древняя, и происходит, вероятно, из Индии. Об этом свидетельствует легенда об изобретателе шахмат, который скромно попросил (после настояний магараджи, которому очень понравилась игра) себе в награду положить одно зерно на угловую клетку шахматной доски и удваивать количество зёрен на каждой следующей клетке. Магараджа, подивившись скудоумию казавшегося таким мудрым человека, распорядился отсыпать ему запрошенные несколько мешков зерна.

5. Оцените приблизительно, во сколько миллионов тонн зерна обойдётся магарадже его щедрость.

Из сказанного выше видно, что если бы на каждое поле шахматной доски не всегда класть столько зерна, сколько просил мудрец, а иногда вообще не класть зёрен, то можно получить таким образом любое число от 0 до $2^{64}-1$. Поэтому, вероятно, таким образом можно представить любое число, которое может встретиться в каких либо конкретных прикладных вычислениях.

Индийская легенда обращает наше внимание на одну особенность двоичной (и любой позиционной) системы — возможность представить колоссальные числа в виде короткой записи. Разумеется, в качестве такой записи не надо использовать совокупность количеств зёрен, лежащих на клетках доски в точности так как указано выше — ведь эти числа могут быть очень велики, и реально такое количество зёрен на большей части клеток доски поместится не может. Вместо этого, как и принято в двоичной системе, на каждую клетку или не кладётся зёрен вообще, или кладётся одно зерно, которое символизирует соответствующую степень двойки. Тогда шахматная доска превращается по существу в то, что на Востоке называют абак, а в России — счёты.

Конечно, реально используемые счёты всегда были десятичными, но проведённые выше рассуждения показывают, что, хотя двоичная запись в три раза длиннее десятичной (и вообще, из всех позиционных систем в этом смысле двоичная — самая плохая),

*) Иногда это приходится делать и в реальной жизни. Различные алгоритмы такого перевода будет изложены далее.

но изготовление счёт с применением двоичной системы могло бы дать определённую (правда, лишь теоретическую) экономию (см. приложение, с. 49).

6. Пусть на каждой из n спиц счётов находится по b костяшек (т. е. счёты представляют числа в системе счисления с основанием $b+1$) и поэтому они позволяют записать в этой системе любое число от 0 до $N=(b+1)^n-1$ (число N характеризует «вместимость» счётов). Каким нужно выбрать b, чтобы суммарное количество костяшек на счётах («сложность» счётов) было минимальным при условии возможности указанного представления на счётах любого числа от 1 до N (т. е. при заданной вместимости)?

Для прочитавших этот параграф, ответ, конечно очевиден. Для знающих логарифмы продолжение этой задачи: сравните сложности десятичных и двоичных счёт одинаковой вместимости.

Приведённый выше алгоритм перевода из десятичной системы в двоичную вычислял цифры двоичной записи, начиная со старших цифр. Опишем теперь кратко, возможно более удобный, алгоритм, в котором цифры двоичной записи вычисляются, начиная с младших*).

Очевидно, самая младшая цифра равна нулю, если число чётное, и единице, если оно нечётное. Для нахождения остальных двоичных цифр надо от исходного числа отнять найденную младшую цифру, поделить разность пополам и к полученному числу применить описанный выше шаг алгоритма.

Например, у числа 300 последние две цифры нули, а для нахождения остальных цифр надо иметь дело с числом $300/4=75$, поэтому следующая цифра 1, и получаем промежуточный результат 37. Следующая далее цифра опять 1, и промежуточный результат 18, поэтому следующая цифра 0, а промежуточный результат 9, следующая цифра 1, а потом три нуля подряд, а старший разряд, как всегда 1. В результате получается двоичная запись 1000101100.

Преимущество этого алгоритма в том, что не требуется предварительного вычисления степеней двойки, но зато приходится неоднократно выполнять операцию деления пополам.

§ 2. ВЗВЕШИВАНИЕ С ПОМОЩЬЮ ГИРЬ И ВОЗВЕДЕНИЕ В СТЕПЕНЬ

Предлагаем читателю самому убедиться в том, что точно так же, как и предыдущем разделе, можно доказать, что для отвешивания любого числа граммов песка от 1 г до n г за одно взвешивание, достаточно иметь гири 1 г, 2 г, 4 г, ..., 2^m г, где $2^m \leqslant n < 2^{m+1}$, и меньшего числа гирь недостаточно, если песок лежит на одной чашке весов, а гири разрешается ставить на вторую чашку. На самом

*) Кстати, кассиры в магазинах и на рынках предпочитают выдавать сдачу начиная с мелких купюр, вопреки инструкции. Причина понятная — надеются, что покупатель, получив мелочь, уйдёт, забыв взять крупные.

деле с математической точки зрения эта задача, известная со средневековых времён, ничем не отличается от рассмотренной выше задачи о конвертах с деньгами.

Часто новые и интересные задачи получаются, если в старой задаче наложить какие-нибудь естественные ограничения. Например, можно задать следующий вопрос: за какое наименьшее количество взвешиваний на чашечных весах можно отвесить килограмм сахарного песка, если имеется лишь одна одногра́ммовая гирька?

На первый взгляд кажется, что единственный способ решения этой задачи — отвесить один грамм, положить в эту же чашку гирьку, отвесить в другой чашке два грамма, переложить гирьку в неё и т. д., добавляя по одному грамму, после тысячного взвешивания отмерить наконец-то килограмм.

Но есть и более быстрый способ. Нужно лишь заметить, что если мы научились отвешивать за n взвешиваний m г песка, то, сделав ещё одно взвешивание, можно даже не используя гирьку отвесить ещё m г, и, ссыпав обе порции вместе, получить $2m$ г за $n+1$ взвешивание. А если при этом взвешивании положить на одну из чашек гирьку, то за $n+1$ взвешивание можно отмерить $2m\pm1$ г песка.

Теперь воспользуемся двоичной записью числа 1000. Применяя любой из указанных выше алгоритмов, получаем равенство

$$1000=2^9+2^8+2^7+2^6+2^5+2^3.$$

Так как

$$2^9+2^8+2^7+2^6+2^5+2^3=(((((2+1)2+1)2+1)2+1)2^2+1)2^3,$$

то, последовательно отвешивая 1, $2+1=3$, $2\cdot3+1=7$, $2\cdot7+1=15$, $2\cdot15+1=31$, $2\cdot31=62$, $2\cdot62+1=125$, $2\cdot125=250$, $2\cdot250=500$, получаем на десятом взвешивании $2\cdot500=1000$ г. Девяти взвешиваний не хватит, потому что за два взвешивания можно отмерить массу не более $3=2^2-1$, за три — не более $7=2\cdot3+1=2^3-1$, за четыре — не более $15=2\cdot7+1=2^4-1$, и за девять взвешиваний — не более $511=2^9-1$.

Если нужно отмерить n г песка, то надо записать n в двоичном виде $a_m\ldots a_1$, где $2^{m-1}\leqslant n<2^m$, $a_m=1$, и воспользоваться формулой

$$n=a_m2^{m-1}+\ldots+a_22+a_1=(\ldots((2a_m+a_{m-1})2+a_{m-2})\ldots)2+a_1,$$

последовательно отвешивая по $b_1=a_m$, $b_2=2b_1+a_{m-1}$, $b_3=2b_2+a_{m-2}$, …, $b_m=b_{m-1}2+a_1=n$ г.

В используемой формуле знающие читатели увидят так называемую схему Горнера. К ней мы ещё вернёмся в дальнейшем.

Идея, лежащая в основе этого метода взвешивания, стара как сама математика. Её применяли и древние египтяне, и древние индусы, но, конечно, не для взвешивания, а для умножения. Ведь алгоритм умножения столбиком был придуман не сразу, а до этого умножение сводилось к сложению. Например, чтобы умножить какое-нибудь

число a на 1000, можно, используя только операции сложения последовательно вычислить $a+a+a=3a$, $3a+3a+a=7a$, $7a+7a+a=15a$, $15a+15a+a=31a$, $31a+31a=62a$, $62a+62a+a=125a$, $125a+125a=250a$, $250a+250a=500a$, $500a+500a=1000a$. Такой метод умножения дожил почти до нашего времени, он удобен при вычислениях на счётах. Сейчас он никому не нужен, так как все используют калькуляторы. Но как возвести на калькуляторе число a, например, в тысячную степень, если у него нет специальной операции возведения в произвольную степень? Умножать 999 раз не нужно, а можно применить тот же приём, последовательно вычисляя $a^3=a^2a$, $a^7=(a^3)^2a$, $a^{15}=(a^7)^2a$, $a^{31}=(a^{15})^2a$, $a^{62}=(a^{31})^2$, $a^{125}=(a^{62})^2a$, $a^{250}=(a^{125})^2$, $a^{500}=(a^{250})^2$, $a^{1000}=(a^{500})^2$.

Если вспомнить, что 1000 имеет двоичную запись 1111101000, то можно заметить, что если отбросить старший бит (всегда равный единице), то каждому следующему биту соответствует операция возведения в квадрат, если он нулевой, или, если он ненулевой, возведение в квадрат с последующим умножением на число a — основание степени (т. е. делается две операции). Кстати, число a не нужно каждый раз заново набирать на клавиатуре. Нужно в самом начале вычислений занести его в память калькулятора, и когда нужно, после нажатия кнопки для умножения, просто вызывать его из памяти и потом нажимать кнопку «равно». Таким образом, возведение в квадрат требует двукратного нажатия кнопок, а возведение в квадрат и последующее умножение на основание степени — пятикратного. Для того чтобы не запутаться в операциях, можно перед началом вычислений составить мнемоническое правило. Возведение в квадрат обозначим символом К, а возведение в квадрат и последующее умножение — символом КУ. Тогда, заменяя в двоичной записи единицы (кроме старшей) на КУ, а нули — на К, получим правило КУКУКУКУККУККК, или короче $КУ^4ККУК^3$.

Посчитаем общее число операций умножения в рассмотренном вычислении. Число возведений в квадрат на единицу меньше длины двоичной записи показателя степени, а число умножений общего вида на единицу меньше суммы цифр двоичной записи.

Для любого n обозначим $\lambda(n)$ уменьшенную на единицу длину двоичной записи числа n, а $\nu(n)$ — её сумму цифр (другими словами, число единиц в ней). Тогда в общем случае число операций умножения, использованных в этом методе возведения в степень n, будет равно $\lambda(n)+\nu(n)-1$. Далее будет показано, что меньшим числом операций обойтись нельзя, если только не обновлять содержимое ячейки памяти.

Очевидно, что $\lambda(n)+\nu(n)-1\leqslant 2\lambda(n)$. Те, кто знают логарифмы, сообразят, что $\lambda(n)=\lfloor\log_2 n\rfloor$, где знак $\lfloor x\rfloor$ означает целую часть числа x. Но можно вычислить обе введённые функции даже не упоминая о двоичной записи. Для этого надо воспользоваться

следующими правилами:

$$\nu(1)=1, \quad \nu(2n)=\nu(n), \quad \nu(2n+1)=\nu(n)+1,$$
$$\lambda(1)=0, \quad \lambda(2n)=\lambda(2n+1)=\lambda(n)+1.$$

Однако для доказательства справедливости этих правил полезно, конечно, воспользоваться двоичной системой, после чего они становятся почти очевидными.

Докажем полезное и простое неравенство $\nu(n+1)\leqslant\nu(n)+1$. Оно очевидно превращается в равенство, если n чётно, так как тогда его двоичная запись заканчивается нулём. Если же эта двоичная запись заканчивается k единицами, перед которыми стоит нуль, то двоичная запись числа $n+1$ заканчивается k нулями, перед которыми стоит единица (а старшие биты остаются без изменения, если они есть). Для того, чтобы в этом убедиться, достаточно выполнить прибавление 1 к n в двоичной системе. В обоих рассмотренных случаях $\nu(n+1)\leqslant\nu(n)+1$.

Из доказанного неравенства следует, что

$$\lambda(n+1)+\nu(n+1)\leqslant\lambda(n)+\nu(n)+1.$$

Действительно, если $2^{k-1}<n+1<2^k$, то $\lambda(n+1)=k-1=\lambda(n)$, и из неравенства $\nu(n+1)\leqslant\nu(n)+1$ следует нужная нам оценка. Если же $n+1=2^k$, то $\lambda(n+1)=k=\lambda(n)+1$, $\nu(n+1)=1$, $\nu(n)=k$, откуда следует, что $\lambda(n+1)+\nu(n+1)=k+1\leqslant 2k=\lambda(n)+\nu(n)+1$.

Справедливо также равенство

$$\lambda(2n)+\nu(2n)=\lambda(n)+\nu(n)+1,$$

которое сразу следует из равенств $\nu(2n)=\nu(n)$, $\lambda(2n)=\lambda(n)+1$.

Выше было показано, что число операций умножения, использованных для возведения в степень n на калькуляторе с одной ячейкой памяти, не больше чем $\lambda(n)+\nu(n)-1$. При $n=1$, 2 очевидно, что меньшим числом операций обойтись нельзя. Покажем, что и в общем случае это так, если только не обновлять содержимое ячейки памяти, т. е. кроме возведения в квадрат всегда использовать только умножение на основание степени.

Допустим противное, а именно, что для вычисления указанным образом x^n при некотором n оказалось достаточно $l<\lambda(n)+\nu(n)-1$ операций. Выберем среди таких чисел n наименьшее число и обозначим его также n. Если последняя операция в рассматриваемом вычислении была возведение в квадрат, то n чётно, и для вычисления $x^{n/2}$ достаточно $l-1<\lambda(n)+\nu(n)-2=\lambda(n/2)+\nu(n/2)-1$ операций, поэтому минимальным числом с рассматриваемым свойством не может быть n, что ведёт к противоречию. Аналогично получается противоречие и в случае, когда последней операцией было умножение на x. Действительно, тогда согласно доказанному выше неравенству для вычисления x^{n-1} достаточно $l-1<\lambda(n)+\nu(n)-2\leqslant\lambda(n-1)+\nu(n-1)-1$ операций.

Но если обновлять содержимое ячейки памяти, то указанный выше метод вычисления x^{1000} можно улучшить. Для этого можно применить так называемый метод множителей. Идея этого метода заключается в следующем. Заметим, что если мы умеем возводить в степень n за $l(n)$ операций и возводить в степень m за $l(m)$ операций, то можно после того, как закончено вычисление x^n, занести его в ячейку памяти и далее вычислить $x^{nm}=(x^n)^m$ за $l(m)$ операций, используя тот же метод, что и для вычисления x^m. Тогда общее число операций будет равно $l(nm)=l(n)+l(m)$.

Вычисляя x^5 старым методом за $\lambda(5)+\nu(5)-1=3$ операции (с помощью последовательности x, x^2, $x^4=(x^2)^2$, $x^5=(x^4)x$) и применяя два раза метод множителей, получаем, что $l(125)=3l(5)=9$. Выполняя ещё три возведения в квадрат, получаем $l(1000)=l(125)+3=12$. Старый же метод требовал $\lambda(1000)+\nu(1000)-1=9+6-1=14$ операций.

Читателю может показаться, что мы слишком много внимания уделили такому специальному и не слишком важному вопросу, как быстрое выполнение возведения в степень. Лет тридцать назад это замечание было бы справедливым. Но в середине 1970-х годов почти одновременно и независимо группой американских математиков (У. Диффи, М. Хеллман, Р. Ривест, А. Шамир, П. Адлеман) и группой английских криптографов (К. Кокс, М. Вильямсон, Д. Эллис) были открыты первые алгоритмы криптографии с открытым ключом*). Благодаря этим алгоритмам теперь частные лица могут обмениваться секретной информацией по общедоступным каналам связи (например, по Интернету) без боязни, что их сообщения прочтут не только конкуренты, но и спецслужбы. Возникшее направление в криптографии быстро превратилось в популярную область математических исследований, которой уже посвящены многочисленные журналы и книги. И во многих самых распространённых алгоритмах важную роль играет операция возведения в степень.

§ 3. АДДИТИВНЫЕ ЦЕПОЧКИ И ФЛЯГИ С МОЛОКОМ

Назовём *аддитивной цепочкой* любую начинающуюся с 1 последовательность натуральных чисел $a_0=1$, a_1, …, a_m, в которой каждое число является суммой каких-то двух предыдущих чисел (или удвоением какого-то предыдущего числа). Обозначим $l(n)$ наименьшую длину аддитивной цепочки, заканчивающейся числом n (длиной цепочки $a_0=1$, a_1, …, a_m называем число m).

Например, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 — аддитивная цепочка, 1, 2, 3, 5, 7, 14 — минимальная цепочка для 14,

*) Англичане сделали это раньше, но им, как сотрудникам секретной криптографической службы, было запрещено опубликовать свои результаты в открытой печати.

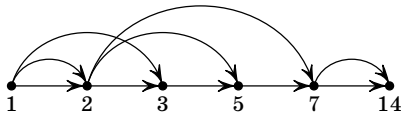


Рис. 1

т. е. $l(14)=5$. Аддитивные цепочки можно изображать в виде ориентированного графа, в котором в вершину a_i идут рёбра от вершин a_j, a_k, если $a_i=a_j+a_k$ (в случае, если такое представление неоднозначно, выбираем любое из них и рисуем только два ребра). Если из какой-то вершины выходит только одно ребро, то для краткости можно «склеить» эту вершину с той вершиной, в которую ведёт это ребро. Граф для предыдущего примера показан на рис. 1.

Можно считать, что все числа в цепочке разные, так как этого легко достичь просто удаляя из неё повторяющиеся числа, и что эти числа расположены в цепочке в порядке возрастания.

Очевидно, что наименьшее число умножений, необходимое для возведения в n-ю степень, равно $l(n)$.

Приведённый выше метод построения аддитивных цепочек называется двоичным (или бинарным). Фактически этим методом было доказано, что справедливо неравенство $l(n)\leqslant\lambda(n)+\nu(n)-1$. Методом множителей легко доказать неравенство $l(nm)\leqslant l(n)+l(m)$.

|| **7.** Докажите нижнюю оценку: $l(n)\geqslant\lambda(n)$.

Из этой оценки следует, что $l(2^n)=n$.

Интересно, что бинарный метод был по существу известен древним индусам, потом был переоткрыт арабскими математиками, задача о точном вычислении функции $l(n)$ появилась в одном французском журнале в 1894 году, потом заново была переоткрыта в 1930-е годы и неоднократно переоткрывалась в дальнейшем, но до сих пор в общем случае не решена.

По существу, наилучшая из известных общих верхних оценок была доказана в 1930-е годы А. Брауэром и имеет вид

$$l(n)\leqslant\lambda(n)\left(1+\frac{1}{\lambda(\lambda(n))}+\frac{C\,\lambda(\lambda(\lambda(n)))}{(\lambda(\lambda(n)))^2}\right),$$

где $C>0$ — некоторая константа.

Не каждую аддитивную цепочку можно вычислить на калькуляторе с одной ячейкой памяти, не используя для запоминания промежуточных результатов собственную голову (фактически, такие калькуляторы имеют две ячейки памяти, так одна из них содержит число, изображаемое в данный момент на дисплее). Укажем, как можно определить необходимое число ячеек памяти для вычисления данной аддитивной цепочки. Для этого введём понятия ширины (а заодно и глубины) аддитивной цепочки.

Пусть дана произвольная цепочка $a_0=1$, a_1, ..., $a_L=n$. Сопоставим каждому её элементу два числа. Первое из них назовём *глубиной элемента*, а второе — *номером ячейки*, хранящей это число. Для элемента a_0 первое число положим равным нулю, а вто-

рое — единице. Будем далее последовательно вычислять эти числа для элементов цепочки. Пусть они уже вычислены для всех элементов от a_0 до a_k. Составим список номеров ячеек, содержащих те элементы цепочки, которые ещё могут быть использованы для вычисления последующих элементов. Найдём наименьшее число, не входящее в этот список, и присвоим его элементу a_{k+1} в качестве номера ячейки (возможно, она использовалась ранее, но теперь уже свободна). Пусть $a_{k+1}=a_i+a_j$, $i, j\leqslant k$. Если $D(a_i)$, $D(a_j)$ — значения глубины элементов a_i, a_j, то положим $D(a_{k+1})$ на единицу большим максимального из чисел $D(a_i)$, $D(a_j)$. *Шириной* S цепочки назовём число использованных ячеек (равное наибольшему из использованных номеров ячеек). *Глубиной* D цепочки назовём глубину её последнего элемента.

8. Докажите, что $a_k\leqslant 2^{D(a_k)}$ и $D\geqslant\lambda(n)$. Докажите, что бинарный метод можно модифицировать так, чтобы длина цепочки не изменилась, а глубина стала бы равна $\lambda(n)$.

Если цепочка имеет ширину S, то её можно представить в виде вычисления на калькуляторе с $S-1$ ячейками памяти (кроме основной, содержащей число, изображаемое в данный момент на дисплее) или в виде компьютерной программы, использующей S ячеек памяти.

Можно ещё представить эту цепочку в виде способа, как налить в данную флягу n литров молока из цистерны, если первоначально в ней был один литр и кроме неё имеется S таких же пустых фляг и весы, способные только сравнивать веса двух фляг между собой. Для этого сопоставим S фляг ячейкам памяти рассматриваемой цепочки, а одну флягу оставим запасной. Тогда любую операцию с ячейками памяти вида $x_k=x_i+x_j$ можно выполнить, выливая в случае необходимости k-ю флягу в цистерну, потом наливая запасную флягу до уровня i-й фляги и сливая её содержимое в k-ю флягу, если $k\neq i$, и делая аналогичную процедуру для индекса j.

Естественно, что аналогичным образом на языке «переливаний» можно представить и программу с командами, использующими не только сложение, но и вычитание $x_k=x_i-x_j$. Поэтому понятие аддитивной цепочки можно обобщить, разрешив использовать вычитание. Для вычисления степеней такие цепочки также можно выполнять на калькуляторе, если кроме умножения использовать и деление. Известно, что в среднем это не даёт существенной выгоды, но в некоторых случаях число используемых операций уменьшается.

Например, вычислить x^{1000} можно с помощью следующей цепочки: 1, 2, 4, 8, 16, 32, 31, 62, 124, 125, 250, 500, 1000.

§ 4. КРАТКАЯ ИСТОРИЯ ДВОИЧНОЙ СИСТЕМЫ

Некоторые идеи, лежащие в основе двоичной системы, по существу были известны в Древнем Китае. Об этом свидетельствует

классическая книга «И-цзин» («Книга Перемен»), о которой речь пойдёт позже.

Идея двоичной системы была известна и древним индусам.

В Европе двоичная система, видимо, появилась уже в новое время. Об этом свидетельствует система объёмных мер, применяемая английскими виноторговцами: два джилла = полуштоф, два полуштофа = пинта, две пинты = кварта, две кварты = потл, два потла = галлон, два галлона = пек, два пека = полубушель, два полубушеля = = бушель, два бушеля = килдеркин, два килдеркина = баррель, два барреля = хогзхед, два хогзхеда = пайп, два пайпа = тан.

Читатели исторических романов, видимо, знакомы с пинтами и квартами. Частично эта система дожила и до нашего времени (нефть и бензин до сих пор меряют галлонами и баррелями).

И в английских мерах веса можно увидеть двоичный принцип. Так, фунт (обычный, не тройский) содержит 16 унций, а унция — 16 дрэмов. Тройский фунт содержит 12 тройских унций. В английских аптекарских мерах веса, однако, унция содержит восемь дрэмов.

Пропагандистом двоичной системы был знаменитый Г. В. Лейбниц (получивший, кстати, от Петра I звание тайного советника). Он отмечал особую простоту алгоритмов арифметических действий в двоичной арифметике в сравнении с другими системами и придавал ей определённый философский смысл. Говорят, что по его предложению была выбита медаль с надписью «Для того, чтобы вывести из ничтожества всё, достаточно единицы». Известный современный математик Т. Данциг о нынешнем положении дел сказал: «Увы! То, что некогда возвышалось как монумент монотеизму, очутилось в чреве компьютера». Причина такой метаморфозы не только уникальная простота таблицы умножения в двоичной системе, но и особенности физических принципов, на основе которых работает элементная база современных ЭВМ (впрочем, за последние 40 лет она неоднократно менялась, но двоичная система и булева алгебра по-прежнему вне конкуренции).

§ 5. ПОЧЕМУ ДВОИЧНАЯ СИСТЕМА УДОБНА?

Главное достоинство двоичной системы — простота алгоритмов сложения, вычитания умножения и деления. Таблица умножения в ней совсем не требует ничего запоминать: ведь любое число, умноженное на нуль равно нулю, а умноженное на единицу равно самому себе. И при этом никаких переносов в следующие разряды, а они есть даже в троичной системе. Таблица деления сводится к двум равенствам $0/1=0$, $1/1=1$, благодаря чему деление столбиком многозначных двоичных чисел делается гораздо проще, чем в десятичной системе, и по-существу сводится к многократному вычитанию.

Таблица сложения как ни странно чуть сложнее, потому что $1+1=10$ и возникает перенос в следующий разряд. В общем виде операцию сложения однобитовых чисел можно записать в виде $x+y=2w+v$, где w, v — биты результата. Внимательно посмотрев на таблицу сложения, можно заметить, что бит переноса w — это просто произведение xy, потому что он равен единице лишь когда x и y равны единице. А вот бит v равен $x+y$, за исключением случая $x=y=1$, когда он равен не 2, а 0. Операцию, с помощью которой по битам x, y вычисляют бит v, называют по-разному. Мы будем использовать для неё название «*сложение по модулю* 2» и символ $\oplus$. Таким образом, сложение битов выполняется фактически не одной, а двумя операциями.

Если отвлечься от технических деталей, то именно с помощью этих операций и выполняются все операции в компьютере.

Для выполнения сложения однобитовых чисел делают обычно даже специальный логический элемент с двумя входами x, y и двумя выходами w, v, как бы составленный из элемента умножения (его часто называют конъюнкцией, чтобы не путать с умножением многозначных чисел) и элемента сложения по модулю 2. Этот элемент часто называют полусумматором.

§ 6. ХАНОЙСКАЯ БАШНЯ, КОД ГРЕЯ И ДВОИЧНЫЙ n-МЕРНЫЙ КУБ

Далее мы рассмотрим несколько интересных задач, в решении которых помогает знание двоичной системы. Начнём мы с задач, в которых используется только одна, самая простая, из лежащих в её основе идей — идея чисто комбинаторная и почти не связанная с арифметикой.

Первая из них — это «Ханойская башня». Головоломку под таким названием придумал французский математик Эдуард Люка в XIX веке.

На столбик нанизаны в порядке убывания размеров n круглых дисков с дырками в центре в виде детской игрушечной пирамидки. Требуется перенести эту пирамидку на другой столбик, пользуясь третьим вспомогательным столбиком (рис. 2). За один ход разрешается переносить со столбика на столбик один диск, но класть больший диск на меньший нельзя. Спрашивается, за какое наименьшее количество ходов это можно сделать. Ответом в этой задаче служит уже известное нам «индийское число» 2^n-1. Люка в своей книге приводит якобы известную

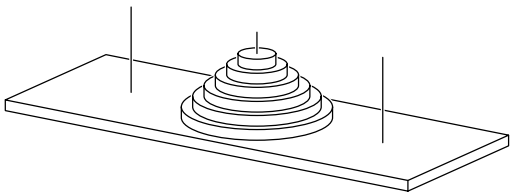


Рис. 2

легенду о том, что монахи в одном из монастырей Ханоя занимаются перенесением на другой столбик пирамидки, состоящей из 64 дисков. Когда они закончат работу, кончится жизнь Брахмы*). Видно, ждать придётся долго.

Решение этой головоломки сильно облегчается, если знать, что такое код Грея. Кодом Грея порядка n называется любая циклическая последовательность всех наборов из нулей и единиц длины n, в которой два соседних набора отличаются ровно в одной компоненте. Примером кода Грея порядка 3 является последовательность трёхразрядных наборов 000, 001, 011, 010, 110, 111, 101, 100.

|| **9.** Докажите, что длина кода Грея порядка n равна 2^n.

Если занумеровать компоненты каждого набора справа налево (при этом последняя, т. е. самая правая компонента получит номер 1), и начинать код Грея с нулевого набора, то его можно записать короче, если вместо очередного набора писать только номер компоненты, в которой он отличается от предшествующего набора. Например, указанный выше код Грея можно коротко записать в виде последовательности семи чисел 1, 2, 1, 3, 1, 2, 1. В общем случае длина подобной последовательности равна 2^n-1. Указанная краткая запись позволяет догадаться, как можно строить коды Грея дальше. Например, Грея порядка 4 можно задать последовательностью 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1. Она получается, если мы повторим два раза последовательность, определяющую код Грея порядка 3, разделив оба экземпляра этой последовательности числом 4. Далее поступаем аналогично, т. е. последовательность длины 2^n-1, определяющую код Грея порядка n, дублируем, разделив оба дубля числом $n+1$. Полученная последовательность длины $2(2^n-1)+1=2^{n+1}-1$ будет определять код Грея порядка $n+1$.

Легко видеть, что эта последовательность, так же, как и предыдущая, не изменяется, если её выписать в обратном порядке. Последовательности (или слова) с таким свойством называются палиндромами. Например, одним из самых известных русских палиндромов является предложение (символы пробела, естественно, игнорируются)

А РОЗА УПАЛА НА ЛАПУ АЗОРА.

Нам понадобится это свойство кода Грея порядка $n+1$, а также то обстоятельство, что все числа этой последовательности, кроме среднего, не больше n. Это означает, что если с её помощью мы начнём выписывать последовательность наборов длины $n+1$, начиная с нулевого, то первые 2^n-1 наборов будут начинаться с нуля, так как $(n+1)$-я компонента никогда в них не будет меняться.

*) Приблизительный западный аналог — это конец света, но на Востоке считают, что после этого рождается новый Брахма, и всё циклически повторяется сначала, от золотого века до нашей Кали-Юги, которой и заканчивается цикл.

Остальные же компоненты будут образовывать наборы длины n, и они будут чередоваться в том же порядке, что и в коде Грея порядка n, поэтому получится некоторая перестановка всех 2^n наборов длины $n+1$, начинающихся с нуля. Потом в её последнем наборе этот нуль будет заменён на единицу (это определяется тем, что в середине последовательности стоит число $n+1$), далее эта единица меняться не будет, а будет меняться в каждом очередном наборе длины $n+1$ только одна из последних n компонент, и эти компоненты образуют код Грея порядка n, выписанный в обратном порядке, и закончится он нулевым набором. Сама же построенная при этом последовательность наборов длины $n+1$ будет образовывать перестановку всех наборов длины $n+1$, начинающихся с единицы, а её последним набором будет набор $10\ldots0$. Таким образом, построенная последовательность состоит из 2^n различных наборов и может быть превращена в циклическую, т. е. является кодом Грея порядка $n+1$.

Код Грея можно наглядно изобразить на n-мерном двоичном кубе. Сам этот куб служит для наглядного представления множества всех наборов длины n из нулей и единиц. Наборы изображаются точками и называются вершинами куба. Два набора, отличающиеся только в одной компоненте, называются соседними и образуют ребро куба. Номер этой компоненты называется направлением ребра. Куб можно нарисовать на плоскости так, что все рёбра будут изображены отрезками, соединяющими их вершины, причём рёбра одного направления будут изображены равными и параллельными отрезками (поэтому такие рёбра называют тоже параллельными). Например, четырёхмерный куб можно изобразить на плоскости так, как показано на рис. 3.

Код Грея на многомерном кубе можно изобразить в виде последовательности вершин, в которой каждые две соседние вершины соединяются рёбрами. Такие последовательности вершин принято называть путями. Но код Грея изображается путём, у которого первая и последняя вершина тоже соединяются ребром. Такие пути естественно называть циклами. Однако код Грея — не просто цикл, а цикл проходящий через все вершины куба. Такие циклы (а их можно искать не только на многомерном кубе, но и на любом графе*)) называются *гамильтоновыми*

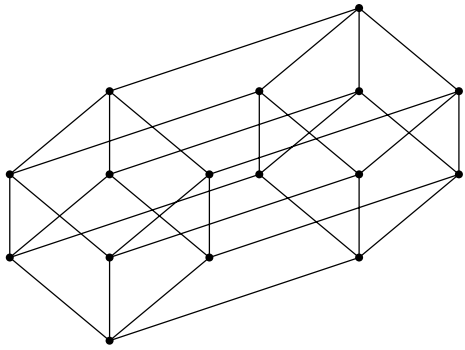


Рис. 3

*) Граф — это произвольное множество вершин, некоторые из которых соединены рёбрами.

*циклами**), а графы, у которых они есть — гамильтоновыми графами. Вопрос о том, какие графы гамильтоновы, а какие нет, оказался чрезвычайно трудным и не решён удовлетворительно по сей день. Ему можно посвятить отдельную книгу, и такие книги уже написаны, поэтому мы прекращаем разговор на эту тему и возвращаемся к многомерному кубу.

Для того, чтобы изобразить код Грея на приведённом на рис. 3 изображении n-мерного куба рядом с вершинами следует написать их имена — соответствующие им наборы из нулей и единиц. Сделать это можно, например, таким образом. Самой нижней вершине сопоставим набор из одних нулей. Рёбрам сопоставим номера их направлений, например, направлению самого правого ребра, выходящего из «нулевой» вершины, сопоставим номер 1, и т. д., а направлению самого левого такого ребра — номер n. Далее сопоставляем оставшимся вершинам куба имена с помощью следующего алгоритма. Если какой-то вершине имя уже присвоено и, поднимаясь из неё вверх по какому-нибудь ребру, скажем, направления k, попадаем в новую, пока безымянную, вершину, то ей присваиваем имя, которое получается из имени прежней вершины заменой k-й компоненты (которая была нулём) на противоположную (т. е. единицу). Если же мы попали в вершину, имя которой уже было присвоено ранее, то можно ничего не делать, так как если мы попробуем всё же сопоставить ей имя согласно указанному правилу, то оно совпадёт с уже присвоенным именем. Очевидно, самой верхней вершине будет присвоен набор из одних единиц. Результат работы описанного алгоритма для четырёхмерного куба показан на рис. 4.

Читателю предоставляется возможность самому решить головоломку Люка и обнаружить её связь с кодом Грея, а мы займёмся другим вопросом.

Бросается в глаза на изображениях многомерного куба, что все вершины, имена которых содержат заданное число единиц, скажем k, лежат на одной прямой (рис. 5). Говорят, что эти вершины лежат на *k-м слое куба*. Очевидно, нулевой слой состоит из одной вершины — «нулевой», а n-й слой состоит только из «единичной» вершины.

10. Сколько различных «возрастающих» путей ведут из «нулевой» вершины в данную вершину k-го слоя?

О т в е т: $k(k-1)(k-2)\cdot\ldots\cdot 2\cdot 1$. Это число называют «$k$ факториал» и кратко обозначают $k!$.

*) В честь ирландского математика, механика, физика и астронома У. Р. Гамильтона, подсказавшего одному книготорговцу идею головоломки — как обойти по рёбрам все вершины правильного многогранника (например, додекаэдра) и вернуться в исходную вершину. Полученные от книготорговца десять фунтов были, вероятно, единственным гонораром знаменитого учёного за многочисленные научные труды, если не считать оклада профессора Дублинского университета.

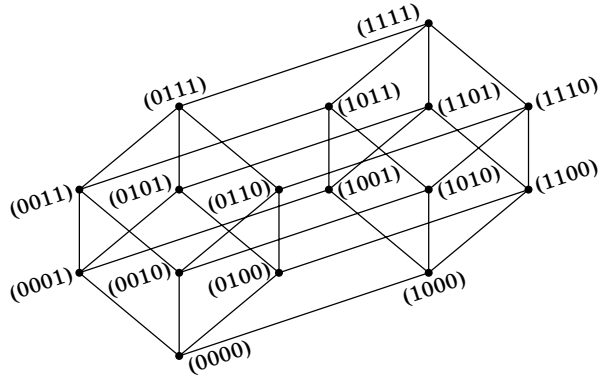


Рис. 4

Для того чтобы решить эту задачу, достаточно «закодировать» любой такой путь последовательностью k различных чисел, нумерующих направления составляющих этот путь рёбер.

В частности, число возрастающих путей от нулевой вершины до единичной равно $n!$. На любом таком пути есть единственная вершина, принадлежащая k-му слою, и она разбивает его на две части. Нижняя часть соединяет её с нулевой вершиной, и этот путь — один из множества возможных путей, соединяющих её с нулевой вершиной, которых, как мы уже знаем, ровно $k!$. Верхняя часть соединяет её с единичной вершиной, и этот путь — один из множества возможных, которых, как легко понять по аналогии, в точности $(n-k)!$.

Поэтому множество всех возрастающих путей от нулевой вершины до единичной, проходящих при этом через заданную вершину k-го слоя, равно $k!\,(n-k)!$. Но всего возрастающих путей от нулевой

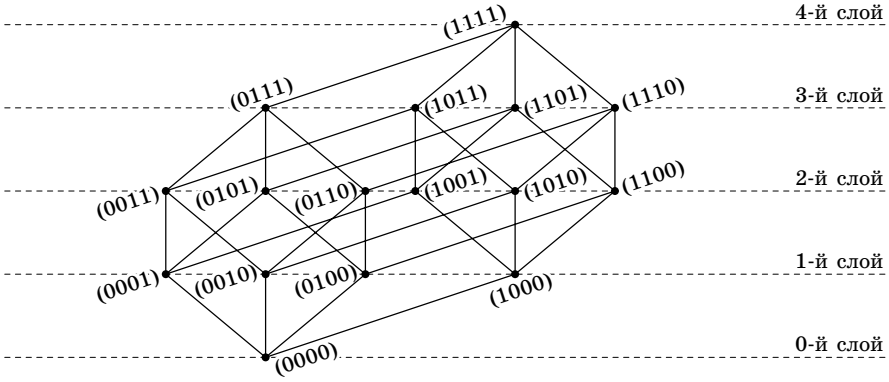


Рис. 5

вершины до единичной $n!$, поэтому число вершин в k-м слое равно $\frac{n!}{k!\,(n-k)!}$. Это число называется *k-м биномиальным коэффициентом* и обозначается кратко C_n^k. Далее оно нам понадобится.

На этом мы заканчиваем рассказ о n-мерном кубе и возвращаемся в заключение опять к коду Грея. Очевидно, что в нём несоседние вершины могут быть соседними на кубе, т. е. соединяться в нём ребром. В некоторых приложениях, как теоретических, так и практических, представляет интерес построение цепи или цикла в n-мерном кубе, в которой несоседние вершины никогда не являются соседними в этом кубе. Такая цепь максимальной возможной длины называется «змеёй в ящике». Какова её длина, до сих пор неизвестно. Наилучшие известные её оценки принадлежат новосибирским математикам А. А. Евдокимову и А. Д. Коршунову.

Другой задачей, связанной с кодом Грея, но более простой, является его недвоичное обобщение. Например, как построить циклическую последовательность всех трёхзначных десятичных чисел от 000 до 999, в которой каждые два соседних числа были бы соседними ещё и в том смысле, что отличались бы ровно в одном разряде и ровно на единицу? Несколько проще построить такую последовательность, если считать цифры 0 и 9 также соседними, хотя разность между ними и не равна 1. Ещё проще построить нециклическую последовательность с теми же свойствами.

Эта задача имеет приложение к алгоритму быстрейшего вскрытия кодового замка на дипломате, поэтому мы не приводим её решения.

Заметим, что этим методом можно вскрыть, конечно, дипломат не только с десятичным, но и с любым k-ичным кодовым замком. Однако при нечётном k аналога циклического кода Грея не существует, а существует только «цепной» код Грея. Читателю предоставляется возможность самому это проверить.

§ 7. КНИГА ПЕРЕМЕН, АЗБУКА МОРЗЕ, ШРИФТ БРАЙЛЯ И АЛФАВИТНЫЕ КОДЫ

Двоичная система, по крайней мере в своей комбинаторной ипостаси, по существу была известна в Древнем Китае. В классической книге «И-цзин» («Книга Перемен») приведены так называемые гексаграммы Фу-си, первая из которых имеет вид  , а последняя (64-я) — вид  , причём они расположены по кругу и занумерованы в точном соответствии с двоичной системой (нулям и единицам соответствуют сплошные и прерывистые линии). Китайцы не поленились придумать для этих диаграмм специальные иероглифы и названия (например, первая из них называлась «кунь», а последняя — «цянь», сплошной линии сопоставляется мужское начало янь, а прерывистой линии — женское начало инь).

Каждая гексаграмма состоит из двух триграмм (верхней и нижней), им тоже соответствуют определённые иероглифы и названия. Например, триграмме из трёх сплошных линий сопоставлен образ-атрибут «небо, творчество», а триграмме из трёх прерывистых линий сопоставлен образ-атрибут «земля, податливость, восприимчивость». Их также принято располагать циклически, но этот цикл не является кодом Грея.

Книга Перемен очень древняя, возможно, одна из древнейших в мире, и кто её написал — неизвестно. Она использовалась ранее, и используется в настоящее время, в том числе и на Западе, для гадания. В Европе с аналогичной целью используются карты Таро. В чём-то обе эти системы схожи, но Таро никак не связаны с двоичной системой, поэтому о них мы говорить не будем.

Способ гадания по Книге Перемен в кратком изложении таков. Бросается шесть раз монета (или лучше пуговица, деньги в гадании применять не рекомендуется) и по полученным результатам (орёл или решка) разыскивается подходящая гексаграмма (для этого надо заранее сопоставить орлу и решке янь или инь). По гексаграмме разыскиваете соответствующий раздел Книги Перемен (имеется перевод выдающегося синолога Ю. К. Шуцкого, неоднократно переиздававшийся в последнее время) и читаете, что там написано.

Конечно, перевод текста книги в предсказание требует опыта и мастерства. И заниматься этим надо после соответствующей подготовки, в подходящем настроении, в подходящее время и в подходящем месте. Говорят, тогда предсказания почти всегда сбываются. А может быть, просто магическим образом из множества вариантов будущего выбирается тот, который соответствует предсказанию?

Заинтересованного читателя отсылаем к книгам Дж. Х. Бреннана «Таинственный И-цзин», М.: Гранд, 2001; В. Фирсова «Книга Перемен. Мистика и магия древнего Китая», М.: Центрполиграф, 2002, и переходим к другой теме — азбуке для слепых.

На этом примере, в частности, хорошо видно, что многие на первый взгляд простые идеи рождались не сразу, и своим появлением обязаны усилиям многих людей. Идею использовать рельефные буквы для печатания книг для слепых первым предложил француз Валентен Ойи. Но выпущенные им книги успехом не пользовались, так как слепым трудно было на ощупь отличать сложные начертания букв друг от друга.

Капитан французской армии Шарль Барбье в 1819 году предложил печатать выпуклыми не буквы, а точки и тире (или просто продавливать их на бумаге) и ими уже записывать буквы. Эту систему он назвал «ночное письмо» и предлагал вовсе не для слепых, а для использования в военно-полевых условиях. С появлением электрических фонариков военное значение этого изобретения упало до нуля.

Слепой мальчик Луи Брайль познакомился с этой системой в 12 лет. Она ему понравилась тем, что позволяла не только читать, но и писать. В течение трёх лет он её усовершенствовал и создал так называемый шрифт Брайля. В нём символы языка (буквы, знаки препинания и цифры) кодируются комбинациями от одной до шести выпуклых точек, расположенных в виде таблицы стандартного размера с тремя строчками и двумя столбцами. Элементы (точки) таблицы нумеруются числами 1, 2, 3 в первом столбце сверху вниз и 4, 5, 6 во втором столбце сверху вниз. Каждая точка либо продавливается специальной машинкой (или даже шилом) или остаётся целой. Всего различных способов продавить выпуклые точки в этой таблице 64 (в том числе и тот, в котором ни одна из точек не вдавлена). При желании теперь читатель может сопоставить каждому символу алфавита Брайля одну из гексаграмм Книги Перемен. Вряд ли, конечно, Брайль знал об этой книге.

Вероятно, не имеет смысла описывать все символы шрифта Брайля, тем более что после его смерти в 1852 году шрифт дополнялся и совершенствовался. Но несколько слов сказать, видимо, стоит. Буква «а», например, изображается одной выпуклой точкой в 1-м элементе таблицы, буква «б» изображается выпуклыми точками в 1-м и 2-м элементах таблицы и т. д. Для сокращения текстов некоторые часто встречающиеся слова или комбинации букв кодируются специальными таблицами. Для того чтобы отличать заглавную букву от строчной, перед ней ставят специальную таблицу, изображающую то, что сейчас называют эскейп-символом. Многие таблицы имеют несколько значений (например, буква и какой-нибудь специальный знак или знак препинания), выбор из которых делается в соответствии с контекстом. Цифры кодируются так же, как и первые буквы алфавита, и, чтобы их отличать, перед последовательностью цифр ставится специальный символ — признак числа, а заканчивается число символом отмены признака числа.

Азбука Брайля по известности уступает азбуке Морзе, хотя и применяется до сих пор в отличие от последней. Сэмюэль Морзе известен однако не только изобретением азбуки. Он был и художником-портретистом (его картина «Генерал Лафайет» до сих пор висит в нью-йоркском Сити-Холле*)), и одним из первых фотографов в Америке (учился делать дагерротипные фотографии у самого Луи Дагерра), и политиком (он баллотировался в 1836 году на пост мэра Нью-Йорка), но самое главное его достижение — изобретение телеграфа (а азбука Морзе понадобилась ему для использования телеграфа). Заодно он изобрёл устройство, которое называется реле. Именно из реле спустя сто лет после Морзе были построены первые компьютеры.

*) Известна также его картина «Человек в предсмертной агонии», после просмотра которой его приятель, известный врач, сказал: «По-моему, малярия».

Начал свои работы в этом направлении он в 1832 году, запатентовал своё изобретение в 1836 году, но публичная демонстрация телеграфа произошла только 24 мая 1844 года. По телеграфной линии, соединяющей Вашингтон с Балтимором, была успешно передана фраза из Библии.

Точки и тире оказались самыми элементарными символами, которые мог передавать его телеграф. Они соответствовали коротким и длинным импульсам электрического тока, передаваемым по телеграфным проводам. Длина импульса определялась нажатием руки телеграфиста на ключ телеграфа. Приём сигнала осуществляло реле, которое после появления в нём импульса тока включало электромагнит, который либо заставлял стучать молоточек, либо прижимал колёсико с красящей лентой к бумажной ленте, на которой отпечатывалась либо точка, либо тире в зависимости от длины импульса.

Азбука Морзе сопоставляет каждой букве алфавита последовательность из точек и тире. Естественней всего использовать такие последовательности длины 6, их всего 64 и хватит даже на русский алфавит. Но Морзе понимал, что длину сообщения желательно уменьшить, насколько возможно, поэтому он решил использовать последовательности длины не более 4, их всего $2+4+8+16=30$. В русском алфавите пришлось не использовать буквы «э» и «ё» и отождествить мягкий и твёрдый знаки. Кроме того, наиболее часто используемым буквам он предложил давать самые короткие коды, чтобы уменьшить среднюю длину передаваемого сообщения. Эту идею в наше время используют с той же целью в алфавитном кодировании.

Здесь имеет смысл ввести терминологию теории кодирования. Определение алфавитного кодирования очень просто. Пусть, например, кодирующим алфавитом является двухбуквенный алфавит, например, состоящий из символов 0, 1. Схемой алфавитного кодирования называется отображение каждой буквы кодируемого алфавита в некоторое слово в кодирующем алфавите (называемое элементарным кодом), в рассматриваемом случае — последовательность нулей или единиц. Пользуясь этой схемой, можно закодировать любое слово в кодируемом алфавите, заменяя в нём каждую букву на соответствующий ей элементарный код, и превратить исходное слово в более длинное слово в кодирующем алфавите. Таким образом, и код Брайля, и азбука Морзе являются алфавитными кодами.

Удобнее всего задать код Морзе в виде четырёхярусного двоичного дерева. Из корня дерева выходят два ребра, из которых правому соответствует тире, а левому — точка. Это — рёбра первого яруса. Из их концов тоже аналогичным образом выходят по два ребра. Это — рёбра второго яруса. Дерево рисуем до четвёртого

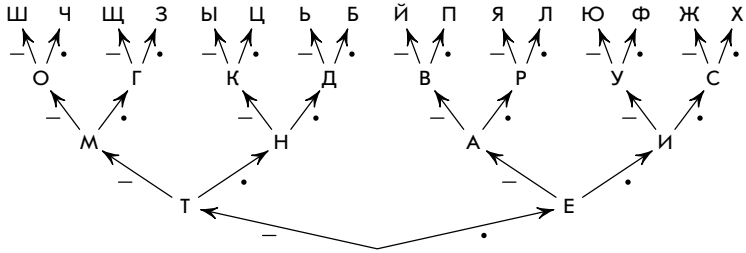


Рис. 6

яруса. Вершинам дерева (за исключением корня) приписываем буквы алфавита (рис. 6). Тогда каждой букве можно сопоставить последовательность точек или тире, получающуюся, если выписать друг за другом последовательность символов, сопоставленных рёбрам дерева, образующим путь, идущий из корня к вершине дерева, соответствующей данной букве.

Очевидно, что алфавитный код должен обладать свойством однозначной декодируемости, т. е. разные слова в исходном алфавите не могут иметь одинаковые коды. А так как процедуру декодирования можно представлять как поиск разбиения закодированного слова на элементарные коды, то это разделение должно быть однозначным. Поэтому однозначно декодируемые коды иногда называют разделимыми кодами. Ясно, что если все элементарные коды имеют одинаковую длину, то код разделим и алгоритм декодирования очень прост. Но если это не так, то код может и не быть разделимым. Например, таким является код Морзе. Но между словами телеграфисты всегда делали промежутки, поэтому никаких проблем не возникало. Однако если промежутки между словами выделять невозможно, приходится использовать только разделимые коды. А пустую букву, изображающую промежуток между словами, часто включают в состав алфавита, что даёт возможность всегда иметь дело не с предложениями, а только со словами.

Понять по достаточно сложному алфавитному коду, является ли он разделимым, бывает не просто. Известны несколько разных алгоритмов для проверки разделимости кода. Наиболее наглядный из них принадлежит А. А. Маркову*).

Не вдаваясь в подробности, ограничимся замечанием, что код, у которого ни один из элементарных кодов не является началом другого (такие коды принято называть *префиксными*), несомненно является разделимым. Предоставим доказательство этой теоремы читателю. Очевидным примером префиксного кода является лю-

*) Заинтересовавшегося этими вопросами читателя мы отсылаем к книге А. А. Маркова «Теория алгорифмов», М.: Наука, 1984; М.: Фазис, 1996.

бой код с равными длинами кодовых слов. Код Морзе префиксным не является, что также предоставляется проверить читателю. Любой двоичный префиксный код можно задать подобно коду Морзе с помощью двоичного дерева, не обязательно такого равномерного, как у него, но при этом буквы кодируемого алфавита должны сопоставляться только «листьям» дерева, но никак не внутренним вершинам.

Возвращаясь к истории алфавитного кодирования, заметим, что его корни уходят в глубь веков. Фактически первый пример применения алфавитного кодирования был описан древнегреческим историком Полибием. Алфавит записывался в квадратную таблицу 5×5 и каждая буква шифровалась парой своих координат (i, j) (номерами строки и столбца), а передаваться сообщения могли в то время с помощью факелов — i факелов в левой руке и j факелов в правой означали пару (i, j).

Дальнейшее развитие идеи алфавитного кодирования принадлежит знаменитому английскому философу, эзотерику и писателю сэру Френсису Бэкону, который первым начал использовать двоичный алфавит в качестве шифроалфавита. В криптографии, правда, это не нашло особого применения, главным образом из-за пятикратного удлинения шифртекста в сравнении с открытым текстом. Но сам Бэкон предложил использовать его как метод, сочетающий криптографию со стеганографией (так называется скрытие самого факта передачи секретного сообщения).

Вместо двоичных цифр он использовал обычный алфавит, но со шрифтами двух типов. Таким методом можно было в любом тексте спрятать шифровку, если, конечно, шрифты были достаточно мало различимы. Желательно при этом использовать разделимый код. Длина зашифрованного сообщения будет в несколько раз короче, чем длина содержащего его (и одновременно маскирующего его) текста, но если для передачи шифровки использовать книгу, то в ней можно таким образом незаметно разместить ещё целую книгу. Но эта красивая идея из-за дороговизны её реализации так и не нашла применения. В наше же время её нельзя рассматривать как серьёзный метод.

Интересно, что в XIX веке, главным образом в кругах, интересующихся наследием возникшего в средневековье тайного мистического ордена розенкрейцеров, появилась идея, что Френсис Бэкон, которого считали розенкрейцером, является настоящим автором пьес Шекспира. Начали искать подтверждение этого в шифрах, которые мог оставить Бэкон в своих книгах, а также в первом знаменитом издании пьес Шекспира. Было, естественно, найдено много таких, якобы зашифрованных фрагментов. Серьёзные исследователи, правда, замечали, что в любом длинном тексте можно при желании и некоторых натяжках найти короткие фрагменты,

напоминающие шифры. Но у сторонников авторства Бэкона стремление доказать это криптографическим методом приняло форму мании. Американский миллионер Фабиан даже создал в начале XX века на свои деньги лабораторию криптоанализа, которая занималась только подобными исследованиями.

Фабиан нанял на работу дипломированного генетика Уильяма Фридмана, сына эмигрантов из России. Через некоторое время Фридман уже возглавлял у Фабиана и лабораторию генетики, и лабораторию криптоанализа. Доказать авторство Бэкона он не смог, более того, он впоследствии опубликовал книгу, где опровергал возможность такого криптографического доказательства. Но он не на шутку увлёкся криптографией и своей подчинённой Элизабет Смит, с которой обвенчался в 1917 году. Они стали самой знаменитой супружеской парой в истории криптографии. После вступления Америки в войну у него с супругой появилась серьёзная работа по правительственным заказам. После войны он ушёл от Фабиана, и стал главным криптографом войск связи.

§ 8. ФОТОПЛЁНКА И ШТРИХ-КОД

Рассмотрим теперь некоторые примеры реального применения двоичного кодирования в современной технике.

Как автоматические фотоаппараты узнают светочувствительность заправленной в них плёнки? Её измеряют в некоторых единицах, и вся выпускаемая сейчас в мире плёнка имеет одно из 24 стандартных значений светочувствительности. Эти значения кодируются некоторым стандартным образом наборами из нулей и единиц, естественно, длины 5. На поверхности кассеты для плёнки нанесены 12 квадратиков чёрного или серебристого цвета, образующих прямоугольник 2×6. Квадратики его верхней части мысленно занумеруем от 1 до 6, начиная слева. Квадратики нижней части аналогично занумеруем от 7 до 12. Серебристые квадратики — это просто металлическая поверхность кассеты, она проводит ток, который с контакта внутри аппарата подаётся на первый квадрат (он всегда серебристый). Чёрные квадраты покрыты краской, не проводящей ток.

Когда плёнка вставляется в аппарат, шесть его контактов соприкасаются с шестью первыми квадратиками, и с квадратиков со 2-го по 6-й снимается информация — нуль, если квадратик чёрный и ток по соответствующему контакту не идёт, и единица в противном случае. Вся информация о светочувствительности плёнки заключена в квадратиках со 2-го по 6-й. В остальных квадратиках заключена информация о числе кадров в плёнке и т. п.

Ещё на поверхности кассеты можно увидеть штрих-код. Это так называемый универсальный код продукта, он сейчас ставится на всех продаваемых товарах. Для чего он нужен и как его прочитать?

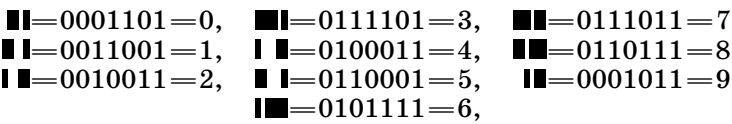
Нужен он только для автоматического занесения информации в кассовый аппарат. Сам штрих-код состоит из тридцати чёрных полос переменной толщины, разделённой промежутками тоже переменной толщины. Толщина полос может принимать четыре значения от самой тонкой до самой толстой. Такую же толщину могут иметь и промежутки. Когда по сканеру проводят штрих-кодом, он воспринимает каждую чёрную полоску как последовательность единиц длины от одной до четырёх, и также воспринимает промежутки между полосами, но при этом вместо единиц сканер видит нули. Полностью весь штрих-код сканер воспринимает как последовательность из 95 цифр 0 или 1 (их давно уже принято называть битами). Что же содержит этот код? Он кодирует 13-разрядное десятичное число, совершенно открыто написанное под самим штрих-кодом. Если сканер не смог распознать штрих-код, то это число кассир вводит в аппарат вручную. Штрих-код нужен лишь для облегчения распознавания сканером изображения. Распознавать цифры, к тому же повёрнутые боком, может только сложная программа распознавания на универсальном компьютере, да и то не очень надёжно, а не кассовый аппарат.

Какую же информацию содержит это 13-значное число? Этот вопрос к математике никакого отношения не имеет. Первая цифра задаёт тип товара, например, у товаров переменного веса она равна 2. Следующие пять цифр — это код производителя, а следующие пять цифр — код самого продукта в принятой этим производителем кодировке. Последняя цифра — это код проверки. Он однозначно вычисляется по предыдущим 12 цифрам следующим образом. Нужно сложить все цифры с нечётными номерами, утроить сумму, к ней прибавить сумму оставшихся цифр, а полученный результат вычесть из ближайшего (большего) кратного 10 числа.

А вот 95-битный код, соответствующий штрих-коду, более интересен. Он содержит в себе только указанное 12-значное число (контрольная цифра в самом штрих-коде не содержится), но с большой избыточностью. Первые три бита в нём, так же, как и последние — это всегда 101. Они нужны только для того, чтобы сканер смог определить ширину полосы, соответствующей одному биту (ведь размеры штрих-кода на разных упаковках могут быть разными) и настроиться на распознавание. В центре кода всегда стоит комбинация 01010, а левая и правая части кода состоят каждая из шести блоков по семь битов и содержат информацию о левых шести и правых шести из данных 12 десятичных цифр. Центральная комбинация позволяет, в частности, отличать поддельные или плохо напечатанные коды.

Цифры 13-значного кода кодируются в левой и правой частях штрих-кода по-разному. В левой половине каждая цифра кодируется семёркой битов, начинающейся с 0 и заканчивающейся 1

согласно следующей таблице:



В правой половине каждая цифра кодируется семёркой битов, начинающейся с 1 и заканчивающейся 0 согласно таблице, которая получается из вышеприведённой, если в ней нули заменить на единицы и единицы на нули (это переход к дополнительному коду). Можно заметить, что каждый из кодов в таблице содержит нечётное число единиц и ровно две группы рядом стоящих единиц и ровно две группы рядом стоящих нулей. Это означает, что каждая цифра соответствует двум соседним полосам на штрих-коде. Но более важно то обстоятельство, что все десять кодов таблицы, будучи прочитанными не слева направо, а справа налево, будут отличаться от любого из кодов таблицы, прочитанного правильным образом. Очевидно, таблица для правой половины кода обладает теми же свойствами, только число единиц в каждом коде чётное.

Такая избыточная (не четырёхбитовая, а семибитовая) таблица кодов нужна для того, чтобы сканер мог правильно прочитать штрих-код и в случае, когда код направляют в него «вверх ногами». Как сканер может отличать одно направление от другого? По чётности или нечётности числа единиц в первом же прочитанном семибитовом блоке, идущем после комбинации 101. При правильном направлении оно будет нечётным, а при обратном направлении — чётным. Перепутать же коды, прочитанные слева, и коды, прочитанные справа, согласно свойству таблицы, невозможно.

Если же в каком-то из семибитовых блоков нарушено правильное чередование нулей и единиц в первом и последнем битах или ему не соответствует чётность числа единиц, то штрих-код признаётся поддельным или плохо пропечатанным.

§ 9. ЗАДАЧИ О ПЕРЕЛИВАНИЯХ

На одной из Всесоюзных математических олимпиад была предложена следующая задача. В три сосуда налито по целому числу литров воды. В любой сосуд разрешается перелить столько воды, сколько в нём уже содержится, из любого другого сосуда. Каждый из сосудов может вместить всю имеющуюся в них воду. Докажите, что можно несколькими переливаниями освободить один из сосудов.

В её решении неожиданно на первый взгляд применяется двоичная система. Так как задача оказалась очень трудной (на олимпиаде её никто не решил), мы приведём здесь это решение. В нём используются две идеи. Первая из них заключается в том, что если будет найден алгоритм переливания, после применения которого ми-

нимальный объём воды, содержащейся в одном из сосудов, уменьшается, то, повторяя многократно этот алгоритм, мы опорожним один из сосудов. Эта идея не такая простая, как может показаться. Ведь это не что иное, как метод бесконечного спуска П. Ферма.

Идея применения двоичной системы лежит в основе этого алгоритма уменьшения минимума. Пусть в сосудах A, B, C находится $a \leqslant b \leqslant c$ литров воды. Разделим b на a с остатком, $b = aq + r$, $0 \leqslant r < a$ и предложим алгоритм, после применения которого в сосуде B останется r литров. Для этого представим q в виде суммы различных степеней двойки: $q = 2^{d_1} + \ldots + 2^{d_k}$. Выливая из сосуда C воду d_1 раз подряд в сосуд A, получим в нём $2^{d_1}a$ литров, а выливая после этого в него $2^{d_1}a$ литров из сосуда B, получаем в нём $b - 2^{d_1}a = a(q - 2^{d_1}) + r = a(2^{d_2} + \ldots + 2^{d_k}) + r$ литров. Аналогично, выливая из сосуда C воду $d_2 - d_1 - 1$ раз подряд в сосуд A, а потом выливая в него воду из сосуда B, получаем в нём $a(2^{d_2} + \ldots + 2^{d_k}) + r - 2^{d_2}a = a(2^{d_3} + \ldots + 2^{d_k}) + r$ литров. Повторяя эту процедуру, получим, что в конце концов в сосуде B окажется r литров воды. Нужно ещё заметить, что во время каждой процедуры из сосуда C выливалось меньше воды, чем из сосуда B, так как $2 + 2^2 + \ldots + 2^{l-1} < 2^l$. Поэтому всего из сосуда C вылито воды меньше, чем из B, значит оба они не опорожнятся раньше времени, и алгоритм работает корректно.

Приведённую выше задачу можно обобщить и на большее количество сосудов. Применив к любым трём из них указанный алгоритм, один из сосудов опорожним. Повторяя эту процедуру ещё раз, опорожним ещё один сосуд и т. д., пока не останутся заполненными только два сосуда.

Предоставляем читателю самостоятельно выяснить, что будет происходить, если продолжить переливания с двумя оставшимися сосудами.

Выше указывалось, что при решении некоторых задач о переливаниях можно использовать аддитивные цепочки. Предлагаем читателю для самостоятельного решения одну из таких задач.

11. Как быстрее всего наполнить флягу 85 литрами молока, пользуясь однолитровым черпаком, если есть ещё одна такая же фляга и весы, способные только сравнивать массы фляг?

Классические задачи о переливаниях выглядят несколько подругому, и метод их решения не связан с двоичной системой.

Примером такой задачи, решение которой по преданию послужило знаменитому французскому математику Пуассону толчком к выбору его профессии, является вопрос о том, как, имея полные сосуды в 3 и 5 литров и пустой 8-литровый сосуд, отмерить ровно 4 литра. Читателю предоставляется возможность самостоятельно придумать метод решения подобных задач за наименьшее количество переливаний.

§ 10. ИГРА «НИМ»

Двоичная система находит неожиданное применение при анализе известной игры «Ним». Происхождение её, так же, как и шахмат, покрыто туманом. Возможно, она была изобретена в Китае.

Состоит она в следующем: на столе лежит несколько кучек спичек, и два игрока по очереди выбирают одну из кучек и забирают из неё сколько угодно спичек (хоть все); выигрывает тот, кто забирает последнюю (есть вариант игры, в котором забравший последнюю проигрывает). Эпизод с этой игрой неоднократно повторяется в известном французском фильме «Прошлым летом в Мариенбаде».

Игра «Ним» являлась излюбленной темой математических кружков в МГУ. Иногда она представлялась в виде гонки нескольких пешек от одного края доски доски до другого. Читатель сам сможет сформулировать правила игры в таком её представлении.

При игре с одной кучкой, очевидно, побеждает начинающий.

При игре с двумя кучками начинающий побеждает не всегда.

12. Докажите, что выигрывающей позицией является позиция с двумя равными кучками. Игрок, сумевший после своего хода попасть в такую позицию, всегда сможет выиграть.

В случае трёх и более кучек описание выигрышной позиции не так просто. Алгоритм распознавания выигрышной позиции следующий. Нужно количество спичек в каждой кучке записать в двоичной системе, и вычислить сумму по модулю 2 полученных двоичных наборов (далее для краткости будем называть её ним-суммой). Для этого вначале нужно вычислить покомпонентную сумму этих наборов, т. е. найти сумму всех младших разрядов, потом сумму следующих за ними разрядов (отсутствующие разряды заменяются нулями) и т. д., и записать полученные суммы в виде (возможно, недвоичного) набора, а потом каждую его компоненту заменить на остаток от деления на 2. Если получится набор из одних нулей, то позиция выигрышная.

Например, если в кучках было 3, 7, 12, 17 спичек, то покомпонентно складывать придётся наборы

$$\begin{array}{rr} & 11\ (=3)\\ + & 111\ (=7)\\ & 1100\ (=12)\\ & 10001\ (=17)\\ \hline & 11223 \end{array}$$

Ним-сумма равна 11001, поэтому позиция является проигрышной для того, кто в неё попал после своего хода. Причина в том, что противник может сделать ход, которым он попадёт в позицию с нулевой ним-суммой. Для этого он может оставить в последней кучке число спичек, равное в двоичной записи ним-суммы наборов

10001 и 11001, т. е. 01000. Тогда ним-сумма чисел, образующих новую позицию, будет равна нулевому набору, так как эта сумма будет отличаться от прежней суммы 11001 прибавлением к ней по модулю 2 набора 11001, что даёт в результате, очевидно, нулевой набор. Поскольку 01000=8, из последней кучки надо взять 17—8=9 спичек.

13. Докажите в общем случае, что из позиции с ненулевой ним-суммой за один ход можно попасть в позицию с нулевой ним-суммой, а из позиции с нулевой ним-суммой любой ход ведёт к позиции с ненулевой ним-суммой.

Теперь ясно, что тот, кто первый попал в позицию с нулевой ним-суммой, дальше при любой игре противника при своём ходе опять сможет попасть в такую же позицию, и в конце концов он возьмёт последнюю спичку.

Указанная выигрышная стратегия поддаётся для реализации даже на специализированных машинах. Одна из таких машин была выставлена после войны в Берлине на английской выставке и с успехом конкурировала с находящимся рядом бесплатным пивным залом. Знаменитый английский математик Алан Тьюринг вспоминал о том, как популярность этой машины повысилась ещё больше после победы над тогдашним бундесминистром экономики Л. Эрхардом.

Читателю предоставляем возможность найти выигрышную стратегию при игре ним в поддавки.

Более интересная модификация игры ним получается, если ограничить число спичек, которые можно взять за один раз, например, числом 10. Тогда интерес представляет даже игра с одной кучкой спичек. Эту игру изобрёл в XVII веке французский математик Баше де Мезириак, написавший кстати, одну из первых в Европе книг по занимательной математике. Читатель может попробовать сам придумать для неё выигрышную стратегию.

§ 11. Д. И. МЕНДЕЛЕЕВ И ТРОИЧНАЯ СИСТЕМА

Когда мы рассматривали задачу о взвешивании с помощью гирь, мы предположили, что груз лежит на одной чашке, а гири — на другой. Но если разрешить класть гири на обе чашки весов, то ответ в задаче об оптимальной системе разновесок изменится. Оптимальной теперь будет система из гирек с массами, образованными степенями тройки. Этой задачей интересовался Д. И. Менделеев в бытность свою председателем Российской палаты мер и весов*).

*) Менделеев имел широкие интересы как в чистой, так и в прикладной науке. Он занимался, например, и экономикой, и демографией, известны его исследования об оптимальной концентрации спирта в воде при производстве водки, по просьбе генштаба он раскрыл состав артиллерийского пороха, используемого немецкой армией. Отвечая на один вопрос Менделеева, А. А. Марков написал знаменитую работу об оценке производной многочлена через величину его максимального значения на отрезке.

Оказалось, что частный случай этой задачи был опубликован в книге Баше де Мезириака в XVII веке, а ранее был известен Фибоначчи. Спрашивалось, какое наименьшее число гирь нужно иметь, чтобы можно было взвесить любой груз от 1 до 40 г. Оптимальным оказался набор гирь 1, 3, 9, 27 г. Для того чтобы взвесить груз в n г, надо представить число n в виде суммы $a_0+3a_1+9a_2+27a_3$, где $a_i=0, \pm 1$ ($i=0, 1, 2, 3$). Тогда для его взвешивания достаточно на чашку вместе с грузом положить все гири, массы которых входят в эту сумму со знаком минус, а на противоположную чашку положить все гири, массы которых входят в эту сумму со знаком плюс.

Но как найти такую сумму? Один из возможных способов решения этой задачи основан на сведении её к представлению числа $n+40$ в виде суммы $b_0+3b_1+9b_2+27b_3$, где $b_i=0, 1, 2$ ($i=0, 1, 2, 3$). Мы уже знаем, что эта задача равносильна представлению числа $n+40$ в троичной системе $n+40=(b_0b_1b_2b_3)_3$. Один из алгоритмов её решения заключается в том, что на правую чашку весов кладутся вначале самые тяжёлые гири, потом гири меньшего веса и т. д. Например, этот алгоритм для числа 40 даёт разложение $40=27+9+3+1$. Если мы уравновесили массу $n+40$ г, положив на чашку b_i гирь массы 3^i ($i=0, 1, 2, 3$), то перекладывая на другую чашку по одной гире каждой массы, мы уравновесим n г. На алгебраическом языке это означает, что будет получено равенство $n=a_0+3a_1+9a_2+27a_3$, $a_i=b_i-1$ ($i=0, 1, 2, 3$). Очевидно, что при этом $a_i=0, \pm 1$ ($i=0, 1, 2, 3$). Верно и обратное, а именно, из разложения $n=a_0+3a_1+9a_2+27a_3$, $a_i=0, \pm 1$ ($i=0, 1, 2, 3$) можно получить разложение $n+40=b_0+3b_1+9b_2+27b_3$, $b_i=0, 1, 2$ ($i=0, 1, 2, 3$). Поэтому из известной нам единственности представления $n+40=b_0+3b_1+9b_2+27b_3$, $b_i=0, 1, 2$ ($i=0, 1, 2, 3$), означающей единственность записи данного числа в троичной системе, вытекает единственность представления $n=a_0+3a_1+9a_2+27a_3$, $a_i=0, \pm 1$ ($i=0, 1, 2, 3$), означающая единственность записи данного числа в так называемой *уравновешенной троичной системе*, $n=(a_0a_1a_2a_3)_3$.

Эта система способна составить некоторую конкуренцию двоичной системе как по простоте арифметических алгоритмов, так и по количеству применений в математических задачах. Удобным её свойством является то, что для изменения знака у представляемого числа достаточно изменить знаки у всех его цифр.

В общем виде доказанные выше утверждения можно записать следующим образом.

Любое целое число от $-(3^n-1)/2$ до $(3^n-1)/2$ может быть однозначно представлено в виде $3^{n-1}b_{n-1}+\ldots+3b_1+b_0$, где $b_i=0, \pm 1$. Для того чтобы взвесить любой груз от 1 до $(3^n-1)/2$ г за одно взвешивание, достаточно иметь гири 1, 3, 9, ..., 3^{n-1} г.

Читатель легко докажет всё это самостоятельно, если заметит, что $(3^n-1)/2=1+3+3^2+\ldots+3^{n-1}$, другими словами, троичная запись числа $(3^n-1)/2$ состоит из одних единиц.

Докажем, что меньшего количества гирь недостаточно, и предложенная система для грузов от 1 до $(3^n-1)/2$ г оптимальна. Допустим, что есть система из $n-1$ гирь с массами g_1, ..., g_{n-1}, позволяющая взвесить любой из этих грузов. Это значит, что любое число m от $-(3^n-1)/2$ до $(3^n-1)/2$ можно представить в виде алгебраической суммы $a_1g_1+\ldots+a_{n-1}g_{n-1}$, $a_i=0, \pm1$, ($i=1$, ..., $n-1$). Но таких сумм ровно 3^{n-1}, так как каждое слагаемое входит в неё с одним из трёх возможных коэффициентов. Но 3^{n-1} меньше общего числа различных грузов, равного 3^n.

14. Покажите, что оптимальная система гирь для взвешивания грузов от 1 до $(3^n-1)/2$ определена однозначно.

15. Докажите, что для взвешивания любого груза от 1 до m г, $(3^{n-1}-1)/2<m\leqslant(3^n-1)/2$, наименьшее количество гирь равно n, а в случае $m<(3^n-1)/2$ выбор гирь неоднозначен.

Уравновешенная троичная система, кроме указанного выше свойства, обладает ещё несколькими удобными свойствами. Например, для выполнения округления в этой системе достаточно просто отбросить лишние цифры. В неуравновешенной системе, даже двоичной, округление выполняется не так просто. Так же просто, как и в уравновешенной системе, производится сравнение чисел по величине, но не нужно обращать при этом внимание на знак числа. Кстати, знак числа в этой системе определяется знаком старшей ненулевой цифры, и не нужно использовать специальный знаковый бит, как в двоичной системе. А таблицы сложения, умножения и деления почти так же просты, как и в двоичной системе. Вычитание же просто сводится к сложению со сменой знака у вычитаемого. При записи чисел в этой системе удобно вместо -1 писать $\overline{1}$. Так как таблица умножения и деления совсем просты, приведём только таблицу сложения:

$$1+1=1\overline{1},\quad 1+\overline{1}=00,\quad \overline{1}+\overline{1}=\overline{1}1,$$
$$1+0=01,\quad \overline{1}+0=0\overline{1}.$$

Умножение, как и деление, тоже сводится к перемене знака и сложению. Другим достоинством троичной системы является то, что запись в ней имеет длину на одну треть меньше, чем в двоичной. Видимо, благодаря им троичная уравновешенная система была положена в основу советского компьютера «Сетунь», построенного в конце 1950-х годов.

Однако широкого распространения она не получила, так как всё же элементная база компьютеров, как того времени, так и современных, остаётся двоичной, а битовое представление троичной системы даже длиннее, чем двоичной.

Уравновешенная система может быть рассмотрена и для любого натурального основания, правда, при чётном основании запись в ней перестаёт быть однозначной. Преимуществом уравновешенных систем является то, что в них записываются и отрицательные числа без знака минус перед записью, а также то, что таблица умножения в этих системах в сравнении с обычными примерно в четыре раза короче, как отметил О. Л. Коши.

16. Запишите число $(1234567890)_{10}$ в уравновешенной десятичной системе счисления.

Утверждение следующей задачи на первый взгляд кажется ложным, однако присмотритесь к ней повнимательнее.

17. Докажите, что любое ненулевое целое число имеет единственное знакопеременное двоичное представление $2^{\alpha_0}-2^{\alpha_1}+\ldots+(-1)^k 2^{\alpha_k}$, где $\alpha_0<\alpha_1<\ldots<\alpha_k$.

Десятичную запись чисел можно преобразовать в запись, состоящую из цифр 0, ±1, ±2, ±3, ±4, ±5, заменяя все цифры, большие 5 на их дополнения до 10, взятые с противоположным знаком, и делая при этом единичный перенос в следующий разряд. Сложение и умножение таких записей можно делать так же, как и обычных, только при этом переносы в следующие разряды могут быть отрицательными. Оценки для числа операций при этом не улучшатся, но сами операции в некотором смысле станут проще, так как для их выполнения достаточно помнить таблицу умножения 5×5. Например, перемножим числа 89 и 98. Запишем первое из них как (1 —1 —1), а второе — (1 0 —2). Умножаем столбиком:

$$
\begin{array}{rrrr}
 & & \times\ 1 & -1\ -1 \\
 & & 1 & 0\ -2 \\
\hline
 & & -2 & 2\quad 2 \\
1 & -1 & -1 & \\
\hline
1 & -1 & -3 & 2\quad 2
\end{array}
$$

К счастью, не пришлось делать переносов, но это не труднее, чем в обычном умножении (заметим, что при умножении обычных записей этих чисел переносы чуть ли не в каждом разряде). Осталось перевести ответ в обычную запись: (1 —1 —3 2 2)=8722.

§ 12. ТРОИЧНАЯ СИСТЕМА И ФОКУС ЖЕРГОННА

Троичная система удачно применяется при объяснении следующего фокуса Жергонна (французского математика XIX века). Зритель запоминает одну из 27 карт и выкладывает их в три стопки по девять карт картинками вверх (первая карта идёт в первую стопку, вторая — во вторую, третья — в третью, четвёртая —

в первую и т. д.). Фокуснику сообщается, в какой из стопок задуманная карта, потом стопки складываются в любом из шести возможных порядков (не перетасовывая карты внутри стопок) и раскладываются снова в три стопки, начиная с верхней карты, потом складываются опять и процедура повторяется в третий раз (каждый раз сообщается, в какую из стопок легла запомненная карта). Фокусник каждый раз замечает, куда легла стопка с запомненной картой — в верх (фокусник запоминает символ 0), в середину (символ 1), или в низ колоды (символ 2), и составляет из этих символов трёхзначное число в троичной системе счисления, ставя первый из замеченных символов в младший разряд, следующий символ — во второй и последний символ — в старший разряд. К полученному числу прибавляется единица и отсчитывается такое количество карт, начиная с верхней карты колоды — последняя из отсчитанных карт и есть запомненная зрителем.

18. Объясните фокус Жергонна.

Имеется ещё один вариант фокуса Жергонна, но с другим способом раскладки карт. Колода из 27 карт раскладывается на три стопки в следующем порядке: первая карта — в первую стопку, вторая — во вторую, третья — в третью, четвёртая — опять в третью, пятая — во вторую, шестая — в первую и т. д. Одна из карт запоминается зрителем и указывается стопка, в которой она лежит, и всё повторяется ещё два раза. Способ угадывания тот же самый. Можно показывать тот же фокус и с 21 картой, но тогда надо раскладывать карты самому и стопку с задуманной картой всегда класть в середину колоды.

19. Докажите, что в фокусе с 21 картой после трёх перекладываний задуманная карта окажется точно в середине колоды, т. е. на 11-м месте от любого края.

Приведём ещё одну задачу, при решении которой может пригодиться троичная система.

20. Докажите, что среди чисел от 1 до $3^n - 1$ можно найти 2^n таких, что никакое среди них не является средним арифметическим двух других.

§ 13. НЕМНОГО ОБ ИСТОРИИ ПОЗИЦИОННЫХ СИСТЕМ СЧИСЛЕНИЯ

Ещё средневековые математики Ближнего Востока нашли простой подход к вычислениям с дробными числами — использование десятичных дробей. Десятичная система попала туда, видимо, из Индии, хотя позиционные дроби, правда не десятичные, а шестидесятеричные, были известны ещё в древнем Шумере, а десятичные дроби, по существу, были известны в древнем Китае. Индейцы майя, вероятно, использовали двадцатеричную систему.

Здесь уместно вспомнить, что запись $(b_n \ldots b_0, b_{-1} \ldots b_{-k})_b$ в позиционной системе счисления с основанием b означает число, равное $b_n b^n + \ldots + b_1 b + b_0 + b_{-1} b^{-1} + \ldots + b_{-k} b^{-k}$, где $b_n b^n + \ldots + b_1 b + b_0$ — его целая, а $b_{-1} b^{-1} + \ldots + b_{-k} b^{-k}$ — дробная часть. В западных странах вместо запятой, отделяющей целую часть от дробной, используется точка.

Почему обычно используется десятичная система? Главным образом, в силу традиции (которая, вероятно, основывается на том, что число пальцев на обеих руках равно обычно 10; индейцы майя, возможно, не забыли и про ноги). Как писал Паскаль, десятичная система ничем не лучше систем с другими основаниями. С некоторых точек зрения более удобны другие системы. Так, много поклонников имеет двенадцатеричная система (идущая от счёта дюжинами и гроссами — дюжинами дюжин). Возможно, к их числу относился и Г. Дж. Уэллс (см. его роман «Когда спящий проснётся»). Преимущество этой системы в том, что 12 имеет больше делителей, чем 10, что несколько упрощает деление. С этой точки зрения ещё лучше шестидесятеричная система (но таблица умножения в этой системе вгоняет в дрожь). Остатки от былого распространения этой системы видны в картографии и астрономии, а алгоритм перевода из этой системы в десятичную запаян в любом калькуляторе для научных расчётов (речь идёт о переводе из градусной меры в десятичную и обратно). Кстати, первая запись дробного числа в позиционной системе в Европе была сделана в XIII веке Фибоначчи: корень уравнения $x^3 + 2x^2 + 10x = 20$ он нашёл в виде $1^\circ\ 26'\ 7''\ 42'''$.

Есть поклонники и у восьмеричной и шестнадцатеричной систем. Первую из них хотел вести в Швеции Карл XII (который, возможно, пришёл к этой идее самостоятельно), но ряд обстоятельств помешали этому прогрессивному начинанию (среди них, вероятно, и занятость короля в военных кампаниях, в частности, в России). Преимущество этих систем в том, что легко осуществляется перевод в двоичную систему и обратно.

Основатель теории множеств уроженец Петербурга Георг Кантор предложил рассматривать системы счисления со смешанными основаниями. Запись в таких системах выглядит так:

$$\begin{matrix} a_3,\ a_2,\ a_1,\ a_0;\ a_{-1},\ a_{-2},\ a_{-3}, \\ b_2,\ b_1,\ b_0;\ b_{-1},\ b_{-2},\ b_{-3}, \end{matrix}$$

где b_i — основания, a_i — цифры, $0 \leqslant a_i < b_i$, а означает эта запись число $a_3 b_2 b_1 b_0 + a_2 b_1 b_0 + a_1 b_0 + a_0 + a_{-1}/b_{-1} + a_{-2}/(b_{-2} \cdot b_{-1}) + \ldots$

Частным случаем таких систем является факториальная, которая получается при $b_k = k+2$, $b_{-k} = k+1$. Используя её, можно любое натуральное число представить в виде $a_n n! + \ldots + a_2 2! + a_1 1!$, где $0 \leqslant a_k \leqslant k$.

Системы со смешанными основаниями всем известны из повседневной жизни. Например, «1 неделя 2 дня 3 часа 4 минуты 56 секунд 789 миллисекунд» равно $\begin{matrix}1,&2,&3,&4,&56;&789\\7,&24,&60,&60;&1000\end{matrix}$ секунд.

§ 14. СХЕМА ГОРНЕРА И ПЕРЕВОД ИЗ ОДНОЙ ПОЗИЦИОННОЙ СИСТЕМЫ В ДРУГУЮ

Использованный в бинарном методе (см. § 2) приём вычисления числа по его двоичной записи является примером более общего алгоритма, называемого *схемой Горнера*. Схема Горнера — это алгоритм для вычисления частного и остатка от деления многочлена $p(x)$ на $x-a$. Кратко опишем, как он устроен и как связан с переводом числа из одной системы в другую.

Пусть дан произвольный многочлен $p(x)=u_nx^n+\ldots+u_1x+u_0$. Деление этого многочлена на $x-a$ — это представление его в виде $p(x)=(x-a)h(x)+r$, $h(x)=v_{n-1}x^{n-1}+\ldots+v_1x+v_0$. Непосредственно можно проверить, что коэффициенты частного можно найти по формулам $v_{n-1}=u_n$, $v_{n-2}=u_{n-1}+av_{n-1}$, ..., $v_0=u_1+av_1$, а остаток можно вычислить по формулам

$$r=u_0+av_0=u_0+a(u_1+av_1)=u_0+a(u_1+a(u_2+av_2))=\ldots$$
$$\ldots=u_0+a(u_1+a(\ldots(u_{n-1}+au_n)\ldots)=u_na^n+\ldots+u_1a+u_0=p(a).$$

Этот метод вычисления и называется схемой Горнера. Слово «схема» в названии алгоритма связано с тем, что обычно его выполнение оформляют следующим образом. Сначала рисуют таблицу $2\times(n+2)$. В левой нижней клетке записывают число a, а в верхней строке — коэффициенты u_n, u_{n-1}, ..., u_0 многочлена $p(x)$, при этом левую верхнюю клетку оставляют пустой:

	u_n	u_{n-1}	...	u_0
a			...	

После этого под числом u_n записывают u_n. Далее на каждом шаге последнее записанное число умножают на a, к результату прибавляют число, стоящее справа сверху от последнего записанного числа, и полученную сумму записывают в клетку справа от этого числа:

	u_n	u_{n-1}	...	u_0
a	$v_{n-1}=u_n$	$v_{n-2}=u_na+u_{n-1}$	...	$p(a)$

Число, которое после выполнения алгоритма оказывается записанным в правой нижней клетке, и есть остаток $p(a)$ деления многочлена $p(x)$ на $x-a$. Другие числа v_{n-1}, v_{n-2}, ... нижней строки являются коэффициентами частного.

Например, деление многочлена $p(x)=x^3-2x+3$ на $x-2$ по описанному алгоритму выполняется так:

1)

	1	0	−2	3
2				

2)

	1	0	−2	3
2	1			

3)

	1	0	−2	3
2	1	$1\cdot2+0=2$		

4)

	1	0	−2	3
2	1	2	$2\cdot2-2=2$	

5)

	1	0	−2	3
2	1	2	2	$2\cdot2+3=7$

6)

	1	0	−2	3
2	1	2	2	7

Получаем, что

$$x^3-2x+3=(x-2)(x^2+2x+2)+7.$$

Общее число операций, используемых в этом алгоритме, равно n плюс число ненулевых коэффициентов у многочлена $p(x)$ минус единица.

Схема Горнера была на самом деле применена англичанином Горнером (а ещё раньше итальянцем Руффини) для вычисления коэффициентов многочлена $p(x+c)$ и использовалась для приближённого вычисления корней многочленов*). Мы укажем некоторые другие её применения. Одно из них — быстрый алгоритм перевода из двоичной системы в десятичную, предложенный Соденом в 1953 году.

Сначала переведём число из двоичной системы в восьмеричную. Для этого разбиваем справа налево его цифры на «тройки» (последняя «тройка» на самом деле может быть «двойкой» или даже одной цифрой) и переводим их в восьмеричную систему схемой Горнера (выполняемой устно). Например,

$$(1111110000)_2=(1.111.110.000)_2=(1760)_8.$$

Выполним перевод из восьмеричной системы в десятичную. Пусть $u=(u_n\ldots u_1)_8$. На k-м шаге выполняем над полученной на предыдущем шаге записью в десятичной арифметике действия

$$\overline{u_n\ldots u_{n-k-1}}-2\cdot\overline{u_n\ldots u_{n-k}}=\overline{v_{n+1}\ldots v_{n-k-1}}$$

и получаем запись $\overline{v_{n+1}\ldots v_{n-k-1}.u_{n-k-2}\ldots u_1}$ (старшие разряды могут оказаться нулевыми и в реальных вычислениях участвовать не будут). На $(n-1)$-м шаге получаем десятичную запись числа u.

*) Впрочем, лежащая в её основе идея была известна Ньютону и, может быть, даже до него.

Например,

$$
\begin{array}{l}
\_1.7\,6\,0 \\
\quad\ 2 \\
\hline
\_1\,5.6\,0 \\
\quad\ 3\,0 \\
\hline
\_1\,2\,6.0 \\
\quad\ 2\,5\,2 \\
\hline
\quad\ 1\,0\,0\,8
\end{array}
$$

Алгоритм перевода из десятичной системы в двоичную, предложенный Розье в 1962 году, почти такой же. Сначала переводим в восьмеричную запись. Для этого, пользуясь восьмеричной арифметикой, на k-м шаге выполняем над полученной на предыдущем шаге записью действия:

$$\overline{u_n \ldots u_{n-k-1}} + 2 \cdot \overline{u_n \ldots u_{n-k}} = \overline{v_{n+1} \ldots v_{n-k-1}}$$

и получаем запись $\overline{v_{n+1} \ldots v_{n-k-1} . u_{n-k-2} \ldots u_1}$ (поначалу $(n+1)$-е разряды окажутся нулевыми и в реальных вычислениях участвовать не будут). На $(n-1)$-м шаге получаем восьмеричную запись числа u. Например,

$$
\begin{array}{l}
+1.9\,4\,5 \\
\quad\ 2 \\
\hline
+2\,3.4\,5 \\
\quad\ 4\,6 \\
\hline
+3\,0\,2.5 \\
\quad\ 6\,0\,4 \\
\hline
\quad\ 3\,6\,3\,1
\end{array}
$$

Далее переводим восьмеричное n-значное число в двоичное (вычисляя для каждой восьмеричной цифры двумя делениями на 2 с остатком её двоичную запись).

21. Переведите из десятичной системы в двоичную систему число 12345678987654321.

22. Переведите из двоичной системы в десятичную систему число 10101010101010101.

§ 15. ПРИЗНАКИ ДЕЛИМОСТИ

Рассказывая о системах счисления, нельзя обойти вниманием признаки делимости. Напомним широко известные признаки делимости в случае использования десятичной системы счисления. Простейший из них следующий: остаток от деления некоторого

числа на 2^n равен остатку от деления на 2^n числа, записанного последними его n цифрами. Аналогичный признак справедлив для 5^n и любого числа вида 2^k5^m, где $\max(m, k)=n$. Чуть более сложен в применении признак делимости на 9: сумма цифр данного числа имеет тот же остаток от деления на 9, что и само число. Такой же признак справедлив и для делимости на 3.

Подобный же признак можно предложить и для делимости на число 9...9, состоящее из n девяток: надо разбить испытуемое число на n-разрядные блоки, начиная с младших разрядов, и всех их сложить (блок, образованный старшими разрядами, может быть короче); у полученного числа будет тот же остаток от деления, что и у исходного. Так как 99 делится на 11, то таким способом можно найти и остаток от деления на 11. Учитывая, что 999 делится на 111 и, следовательно, на 37, получаем признаки делимости на эти числа.

Но есть более эффективный признак делимости на 11: надо складывать цифры числа, начиная с младших, чередуя знаки (первая цифра берётся со знаком плюс) — полученное число имеет тот же остаток от деления на 11, что и исходное.

Аналогичный признак делимости имеется и для числа 10...01, запись которого, кроме двух единиц, содержит n нулей. Испытуемое число разбивается на $(n+1)$-разрядные блоки, начиная с младших разрядов (блок, образованный старшими разрядами, может быть короче), и все они складываются с чередующимися знаками (первое число берётся со знаком плюс). Полученный результат имеет тот же остаток от деления, что и испытуемое число. Поскольку $1001=11\cdot7\cdot13$, мы попутно получаем таким путём признаки делимости на 7, 13, 91, 77, 143.

|| **23.** Докажите сформулированные признаки делимости.

При применении рассмотренных признаков к большим числам получаются меньшие, но всё же достаточно большие числа, имеющие те же остатки от деления, что и исходные. К ним нужно применить ещё раз тот же признак делимости и т. д. Часто эффективность этих признаков при применении к большим числам всё же ненамного выше простого деления.

Есть, однако, случаи, когда только применение признаков делимости позволяет найти остаток, так как непосредственное деление практически невозможно ввиду колоссальной вычислительной сложности.

|| **24.** Найдите остаток от деления 44444^{44444} на 99.

Число 44444^{44444} состоит более чем из 200 000 цифр, и его прямое вычисление требует порядка миллиарда операций. Правда, кроме признаков делимости на 9 и 11, здесь надо применить и некоторые соображения, связанные с делимостью степеней. Другой способ решения этой задачи — применение калькулятора и бинарного метода возведения в степень.

Полезны признаки делимости и при разгадывании ребусов, подобных следующему.

25. Замените звёздочки на пропущенные цифры в примере

$$\begin{array}{r} \times\ 792 \\ \hline ****** \\ \hline 70{*}{*}34{*} \end{array}$$

Признаки делимости могут помочь в нахождении ошибки при выполнении умножения больших чисел. Простейший из способов контроля — проверка по модулю 9. Этому способу обучали ещё в средневековых университетах: у обоих множителей находятся остатки от деления на 9, потом исходные числа перемножаются и у результата опять находится остаток от деления на 9, который сравнивается с остатком от деления на 9 произведения исходных чисел, подлежащего проверке. Если остатки разные, то произошла ошибка. Если известно, что ошибка только в одном разряде, то можно точно указать величину ошибки, но не её положение. В случае совпадения результатов остаётся возможность одной ошибки типа замены 0 на 9 или наоборот, а также нескольких ошибок. В этом случае можно провести ещё аналогичную проверку по модулю 11, которая или подтвердит существование ошибки указанного вида, или установит наличие нескольких ошибок (или их отсутствие).

§ 16. АРИФМЕТИЧЕСКИЕ КОДЫ

Можно установить точно положение ошибки и даже исправить её, в предположении, что она только одна. Для этого надо применить так называемые арифметические коды. Приведём их простейший пример.

Допустим, что при перемножении десятичных чисел получилось 15-разрядное число с ошибкой в одном разряде. Для нахождения величины ошибки применим проверку по модулю 9 и найдём, что она по модулю 9 равна a. Если ошибка произошла в i-м разряде*), то величина ошибки в произведении равна $a\cdot 10^{i-1}$ или $(a-9)\cdot 10^{i-1}$, а если $a=0$, то или ошибки не было, или она равна $\pm 9\cdot 10^{i-1}$. Применим проверку по модулю 31. После неё станет известно значение ошибки b по модулю 31. Если остаток по модулю 31 равен нулю, то ошибки не было.

Пусть он не равен нулю. Выпишем остатки от деления чисел $10, \ldots, 10^{15}$ на 31. Они равны 10, 7, 8, 18, 25, 2, 20, 14, 16, 5, 19, 4, 9, 28, 1. Заметим, что невозможно равенство по модулю 31 чисел $a\cdot 10^i$ и $(a-9)\cdot 10^j$, так как тогда при некоторых $a=1, 2, 3, 4$ и $i=0, 1, \ldots, 14$ были бы равны по модулю 31 числа $a\cdot 10^i$ и $a-9$, а невозможность этого проверяется непосредственно

*) Разряды нумеруются с конца десятичной записи. Например, 3-й разряд — это разряд сотен.

с помощью вычисленной выше последовательности остатков. Точно также проверяется невозможность совпадения по модулю 31 чисел $9\cdot10^i$ и $(-9)\cdot10^j$. Вычисляя остатки по модулю 31 у всех чисел $a\cdot10^i$ и $(a-9)\cdot10^i$ при $i=1, \ldots, 15$ и сравнивая их с найденным ранее остатком, находим значение i и точную величину ошибки a или $a-9$. Аналогично поступаем в случае ошибки $\pm9\cdot10^i$.

Приведённый пример арифметического кода иллюстрирует принципиальную возможность их построения, на первый взгляд кажущуюся парадоксальной. Прикладного значения при ручных вычислениях он не имеет хотя бы потому, что, как можно проверить, проще ещё раз перемножить эти числа, чем выполнять указанные выше операции.

Но такие алгоритмы применяют для контроля правильности работы арифметических устройств, и этот контроль осуществляет специальный блок в таком устройстве. В рассматриваемом случае сложность устройства со встроенным арифметическим кодом рассмотренного вида увеличивается не более чем в два раза. Если рассмотреть подобные коды с достаточно большой длиной $(p-1)/2$ и проверочными множителями 9 и p, где p — простое число вида $280k+31$ такое, что $10^{(p-1)/2}$ при делении на p даёт остаток 1, а 10^k при $k<(p-1)/2$ при делении на p дают остатки, отличные от 1 (в рассмотренном случае p равнялось 31, а k — нулю), то сложность исправления ошибки будет мала при больших p в сравнении со сложностью умножения $(p-1)/2$-разрядных чисел.

На самом деле, на практике использовались для повышения надёжности арифметических устройств не десятичные, как рассмотренный, а двоичные коды, но они устроены подобным же образом.

26. Примените указанный арифметический код для расшифровки ребуса

$$\begin{array}{rr} & 9 \\ \times & ** \\ & 31 \\ & ******* \\ \hline & 425021067* \end{array}$$

Можно с помощью этого кода расшифровать и ребус, в котором ровно в одном разряде результата имеется ошибка, но неизвестно в каком. Тот же код можно использовать для демонстрации арифметического фокуса: вы предлагаете вашему другу задумать два не слишком больших числа, перемножить их и результат умножить на якобы случайное число 279, а потом сообщить вам ответ с одной ошибкой в каком-нибудь разряде. Используя указанный код (если перед этим предварительно подготовить все нужные таблицы и немного потренироваться), вы быстро укажете эту ошибку. Заметьте, что полный перебор вариантов требует в случае 15-разрядного ответа 134-кратного деления предполагаемого ответа на 279.

§ 17. МИНИМАЛЬНЫЕ ФОРМЫ ДВОИЧНОЙ ЗАПИСИ С ЦИФРАМИ 0 И ± 1 И ПЕРВАЯ ПОПЫТКА УМЕНЬШИТЬ СЛОЖНОСТЬ УМНОЖЕНИЯ

В позиционных системах счисления с заданным основанием b можно, кроме обычных цифр, использовать и отрицательные цифры -1, -2, ..., $-(b-1)$. Правда, это приводит к неоднозначности в записи чисел. Зато таким образом можно уменьшить количество ненулевых цифр в записи и их величину. Далее в этом параграфе мы будем рассматривать случай $b=2$, т. е. записи чисел в двоичной системе с цифрами -1, 0, 1.

27. Приведите пример числа, для которого существуют по крайней мере две записи описанного вида с минимальным возможным числом ненулевых цифр.

Назовём двоичную запись с использованием отрицательных единиц *минимальной формой*, если в ней нет соседних ненулевых цифр. Для этого определения не очевидна ни единственность минимальной формы, ни минимальность длины минимальной формы. Однако и то, и другое верно, т. е. минимальная форма определена однозначно и содержит наименьшее количество ненулевых цифр среди всех возможных форм двоичной записи числа с использованием отрицательных единиц.

Докажем это. Пусть $A=\overline{a_n \ldots a_0}$ — произвольная двоичная запись числа a, т. е.

$$a=2^n a_n+\ldots+2a_1+a_0,$$

где $a_i=0, \pm 1$. Далее вместо -1 будем писать $\overline{1}$. Обозначим через $\nu(A)$ количество ненулевых цифр в этой записи и через $\mu(A)$ — количество пар соседних ненулевых цифр. Заметим, что

$$2^k+2^{k+1}+\ldots+2^{m-1}=2^m-2^k,$$

поэтому, выполняя в записи A следующие преобразования:

1° $\quad \alpha\overline{\alpha} \to 0\alpha,$

2° $\quad 0\underbrace{\alpha\alpha\ldots\alpha}_{n} \to \alpha\underbrace{0\ldots0}_{n-1}\overline{\alpha} \qquad (n\geqslant 3),$

3° $\quad 0\underbrace{\alpha(\alpha 0)\ldots(\alpha 0)}_{n}\alpha\alpha \to \alpha\underbrace{0(0\overline{\alpha})\ldots(0\overline{\alpha})}_{n+1} \qquad (n\geqslant 1),$

4° $\quad 00\underbrace{(\alpha 0)\ldots(\alpha 0)}_{n}\alpha\alpha \to 0\alpha\underbrace{(0\overline{\alpha})\ldots(0\overline{\alpha})}_{n+1} \qquad (n\geqslant 0),$

5° $\quad \overline{\alpha}0\underbrace{(\alpha 0)\ldots(\alpha 0)}_{n}\alpha\alpha \to \underbrace{(0\overline{\alpha})\ldots(0\overline{\alpha})}_{n+2} \qquad (n\geqslant 0)$

(где для $\alpha=1,\ \overline{1}$, соответственно, $\overline{\alpha}=\overline{1},\ 1$), мы не меняем записываемого числа, не увеличиваем величин $\nu(A)$ и $\mu(A)$ и всегда уменьшаем их сумму:

Операция	μ'	ν'
1°	$\mu-1$ или $\mu-2$	$\nu-1$
2°	$\mu+2-n$ или $\mu+1-n$ $(n\geqslant 3)$	$\nu+2-n$
3°	$\mu-1$ или $\mu-2$	$\nu-1$
4°	$\mu-1$	ν
5°	$\mu-1$ или $\mu-2$	$\nu-1$

Будем выполнять эти преобразования, пока это возможно. Так как величина $\nu(A)+\mu(A)$ не может неограниченно уменьшаться, то в конце концов получим запись числа a, в которой нельзя будет выполнить ни одну из этих операций.

28. Докажите, что если в записи невозможно выполнить ни одну из указанных операций, то в этой записи нигде не будет встречаться соседних ненулевых цифр.

Докажем единственность минимальной формы для данного числа a. Допустим, что есть две разные минимальные формы A и B. Тогда они заканчиваются одинаковым числом нулей в младших разрядах, иначе, если бы одна заканчивалась k нулями, а другая $m>k$ нулями, то наше число делилось бы на 2^m, а с другой стороны, делилось бы только на 2^k, но не на 2^{k+1}, что невозможно. Аналогично получаем, что их последние ненулевые цифры равны, так как в противном случае наше число имело бы при делении на 2^{m+2} (где m — число нулей в конце) в остатке разные числа 2^m и $2^{m+2}-2^m$ (так как в конце одной записи стоят цифры $010\ldots0$, а в конце другой — цифры $0\overline{1}0\ldots0$ ввиду отсутствия пар ненулевых цифр в обеих записях). Отбрасывая равные последние цифры от обеих записей, получаем более короткие различные записи для равных чисел. Повторяя для этих записей проведённое рассуждение, получим, наконец, что число ±1 или 0 имеет две разные записи, а это невозможно.

Наконец, докажем, что в минимальной форме наименьшее количество ненулевых цифр среди всех записей с отрицательными единицами. Действительно, из любой записи можно с помощью рассмотренных преобразований получить построенную запись (как только что доказано, всегда одну и ту же). Но при выполнении этих преобразований величина $\nu(a)$ не возрастала, значит, построенная запись имеет значение этой величины, равное наименьшему возможному.

Преобразование обычной двоичной записи числа a к минимальной форме более удобно проводить следующим образом: вычислить обычную двоичную запись $\overline{\gamma_{n+1}\ldots\gamma_1}$ числа $3a$ и вычислить обычные разности $\delta_{i-1}=\gamma_i-\alpha_i$, где $i=2, \ldots, n+1$, α_i — цифры записи числа a, а $\alpha_n=\alpha_{n+1}=0$, тогда $\overline{\delta_n\ldots\delta_1}$ — минимальная форма числа a.

Минимальная форма максимум на единицу длиннее обычной записи, но содержит не более $n/2$ ненулевых цифр. При смене знака у числа меняются знаки у всех цифр его минимальной формы. Действительно, при замене знаков всех цифр минимальной формы числа a получается запись числа $-a$, в которой нет двух ненулевых цифр подряд. А это по определению и есть минимальная запись числа $-a$.

Одно из возможных применений указанной минимальной формы — уменьшение числа арифметических операций для возведения в степень. Мы уже приводили конкретный пример такого применения, а сейчас сформулируем общую теорему. Далее для краткости вместо словосочетания «число арифметических операций алгоритма» будем писать *«сложность алгоритма»*.

Обозначим число ненулевых цифр в записи числа n в виде минимальной формы через $\nu(n)$, а уменьшенную на единицу длину этой записи — через $\lambda(n)$. Мы используем те же обозначения, что и в обычном бинарном методе (см. § 2), но заметим, что новая функция $\lambda(n)$ может быть на единицу больше старой, зато новая функция $\nu(n)$ не может быть больше старой, а часто меньше её, иногда почти в два раза.

Теорема. При использовании калькулятора с одной ячейкой памяти сложность вычисления x^n не превосходит

$$\nu(n)+\lambda(n)-1.$$

Доказательство. Используя полученную минимальную форму, запишем n в виде суммы $2^{\lambda(n)}\alpha_{\lambda(n)}+\ldots+2\alpha_1+\alpha_0$, $\alpha_i=0, \pm1$, содержащей $\nu(n)$ ненулевых слагаемых. Далее, как и в обычном бинарном методе возведения в степень, используем аналог схемы Горнера, а цифрам -1 сопоставляем операцию деления на основание степени. Полученное обобщение бинарного метода использует не более $\lambda(n)$ возведений в квадрат и $\nu(n)-1$ умножений и делений. Теорема доказана.

Аналогично обычному бинарному методу можно доказать соотношения $\lambda(2n)+\nu(2n)=\lambda(n)+\nu(n)+1$, $\nu(n\pm1)\leqslant\nu(n)+1$. Из последнего неравенства следует, что $\lambda(n\pm1)+\nu(n\pm1)\leqslant\lambda(n)+\nu(n)+1$. Действительно, случай $n\pm1=n+1$ рассматривается аналогично обычному бинарному методу, а в случае $n\pm1=n-1$, очевидно, $\lambda(n-1)\leqslant\lambda(n)$ и из неравенства $\nu(n-1)\leqslant\nu(n)+1$ следует нужная оценка.

Используя доказанное неравенство, можно аналогично обычному бинарному методу в случае, когда содержимое ячейки памяти

никогда не обновляется, получить аналогичную нижнюю оценку сложности возведения в степень $\nu(n)+\lambda(n)-1$. Читателю предоставляется возможность самому убедится в этом.

Недостатком двоичной системы при её ручном использовании является то, что из-за увеличения длины записи по сравнению с десятичной системой соответственно возрастает и сложность умножения. Использование минимальной формы позволяет уменьшить сложность ручного умножения двоичных чисел. Опишем алгоритм умножения, предложенный в начале 1950-х годов американским математиком Бутом.

Для этого данные n-разрядные и m-разрядные двоичные числа преобразуем в их минимальные формы и заметим, что эти формы содержат не более $n+1$ и $m+1$ разрядов, причём из них не более $a=n/2+1$ и $b=m/2+1$ ненулевых разрядов соответственно. Сложность преобразования не превосходит $3n+3m-4$. Умножая минимальные формы с помощью школьного алгоритма, замечаем, что число нетривиальных умножений не превосходит ab, так как ненулевых строк будет не более b и в каждой из них нетривиальных умножений не более a. Отметим, что каждое нетривиальное умножение, по-существу, не сложнее нетривиального умножения в обычной двоичной системе, и будем считать, что оно выполняется с единичной сложностью, так же как и нетривиальное сложение (операция нетривиальна, если оба операнда не нули). Заметим также, что число нетривиальных сложений не превосходит $(b-1)\times$ $\times(a+n-1)$, так как всего сложений различных строк требуется не более $b-1$, а каждое из них состоит из не более чем n переносов (переносы могут быть как 1, так и -1) и не более чем $a-1$ нетривиальных сложений (в складываемых строчках имеется не более $a-1$ стоящих друг под другом ненулевых цифр). Поэтому сложность умножения не превосходит $(b-1)(a+n-1)+ab\leqslant$ $\leqslant mn+(m+n)/2+1$. Полученный результат содержит не более $n+m+2$ разрядов (так как он получается при сложении $m+1$ не более чем $(n+1)$-разрядных чисел с соответствующими сдвигами). Его можно преобразовать к обычной двоичной записи, сделав не более $n+m+2$ операции (заменяем блоки соседних цифр вида $10\ldots0\overline{1}$ на соответствующие блоки вида $01\ldots11$, блоки вида $1\overline{1}$ — на блоки 01, блоки без отрицательных цифр оставляем без изменения). Значит, полная сложность операции умножения не превосходит $mn+(m+n)/2+1+3n+3m-4+n+m+2\leqslant mn+3{,}5(m+n)$.

§ 18. БЫСТРОЕ УМНОЖЕНИЕ МНОГОЧЛЕНОВ

Мало кто знает, что относительно недавно были открыты гораздо более быстрые алгоритмы умножения и деления многозначных чисел и многочленов, чем «школьные». Первый такой

алгоритм придумал в 1962 году А. А. Карацуба, отвечая на вопрос А. Н. Колмогорова. Впоследствии А. Л. Тоомом, Ф. Штрассеном и А. Шенхаге были построены ещё более быстрые алгоритмы.

Идею метода Карацубы можно пояснить на следующем примере. Пусть перемножаются восьмизначные числа $U=\overline{u_1\ldots u_8}$ и $V=\overline{v_1\ldots v_8}$. Представим их как двузначные числа в системе счисления с основанием 10^4: $U=\overline{U_1U_2}$, $V=\overline{V_1V_2}$, где $U_1=\overline{u_1u_2u_3u_4}$, $U_2=\overline{u_5u_6u_7u_8}$, $V_1=\overline{v_1v_2v_3v_4}$, $V_2=\overline{v_5v_6v_7v_8}$. Тогда их произведение можно представить в следующем виде:

$$UV=10^8U_1V_1+10^4((U_1-U_2)(V_2-V_1)+U_1V_1+U_2V_2)+U_2V_2.$$

Эта формула сводит умножение восьмизначных чисел к трём операциям умножения и шести операциям сложения-вычитания четырёхзначных чисел (с учётом переносов в следующие разряды). Обычный способ требует четырёх умножений и трёх сложений-вычитаний, но так как три раза сложить четырёхзначные числа можно быстрее, чем один раз перемножить, то метод Карацубы уже восьмизначные числа перемножает быстрее. В общем случае он требует для перемножения n-значных чисел по порядку не больше $n^{\log_2 3}<n^{1,585}$ операций над цифрами.

Далее мы поговорим о сложности умножения более подробно.

Обозначим через $M(n)$ наименьшее количество операций сложения, вычитания и умножения (выполняемых над коэффициентами многочленов и промежуточными числовыми результатами), требующихся для перемножения двух многочленов степеней, меньших n. Тогда справедливо неравенство

$$M(n)\leqslant 2M(\lceil n/2\rceil)+M(\lfloor n/2\rfloor)+4\lfloor n/2\rfloor+2n-4\ *). \qquad (*)$$

Действительно, применим равенство

$$(f_1x^{\lfloor n/2\rfloor}+f_0)(g_1x^{\lfloor n/2\rfloor}+g_0)=$$
$$=f_1g_1x^{2\lfloor n/2\rfloor}+((f_1+f_0)(g_1+g_0)-f_1g_1-f_0g_0)x^{\lfloor n/2\rfloor}+f_0g_0,$$

где степени многочленов f_1 и g_1 меньше $\lceil n/2\rceil$, а степени многочленов f_0 и g_0 меньше $\lfloor n/2\rfloor$, и заметим, что для вычисления произведений f_1g_1, f_0g_0 требуется не более $M(\lceil n/2\rceil)+M(\lfloor n/2\rfloor)$ операций, для вычисления сумм f_1+f_0, g_1+g_0, $f_1g_1+f_0g_0$ нужно не более $2\lfloor n/2\rfloor+2\lfloor n/2\rfloor-1$ операций (так как число операций равно наименьшему из количеств ненулевых коэффициентов у складываемых многочленов), для вычисления произведения $(f_1+f_0)(g_1+g_0)$ используется не более $M(\lceil n/2\rceil)$ операций, для вычисления разности $(f_1+f_0)(g_1+g_0)-f_1g_1-f_0g_0$ достаточно $n-1$ операций, так как

*) Через $\lfloor x\rfloor$ обозначается наибольшее целое число, не большее x («округление вниз»), а через $\lceil x\rceil$ — наименьшее целое число, не меньшее x («округление вверх»).

$(f_1+f_0)(g_1+g_0)-f_1g_1-f_0g_0=f_1g_0+f_0g_1$, значит, степень этого многочлена равна $\lfloor n/2\rfloor+\lceil n/2\rceil-2=n-2$, сложение многочленов f_0g_0 и $f_1g_1x^{2\lfloor n/2\rfloor}$ выполняется «бесплатно», так как они не имеют подобных членов, причём в их сумме отсутствует член вида $x^{2\lfloor n/2\rfloor-1}$, поэтому для сложения многочленов $f_0g_0+f_1g_1x^{2\lfloor n/2\rfloor}$ и $(f_1g_0+f_0g_1)\times$ $\times x^{\lfloor n/2\rfloor}$ достаточно $n-2$ операций. В итоге требуется дополнительно $4\lfloor n/2\rfloor+2n-4$ операций.

Оценку сложности метода Карацубы можно представить в следующем виде. Если n кратно 2^k, то справедливо неравенство

$$M(n)\leqslant 3^k\left(M\left(\frac{n}{2^k}\right)+\frac{8n}{2^k}-2\right)-8n+2,$$

а при любом n — неравенство

$$M(n)<\frac{35}{3}n^{\log_2 3}.$$

Действительно, пусть $2^km=n$. Тогда неравенство

$$M(n)\leqslant 3^k\left(M\left(\frac{n}{2^k}\right)+\frac{8n}{2^k}-2\right)-8n+2,$$

доказывается индукцией по k. База ($k=1$) — это неравенство (*). Шаг индукции обосновывается тем же неравенством.

Выберем k так, чтобы $2^k<n\leqslant 2^{k+1}$. Тогда, если $3\cdot 2^{k-1}<n$, то

$$M(n)\leqslant M(2^{k+1})<3^{k-1}(M(4)+30)\leqslant 3^{k-1}\cdot 55<55\left(\frac{n}{3}\right)^{\log_2 3}<\frac{35}{3}n^{\log_2 3}.$$

Если же $n\leqslant 3\cdot 2^{k-1}$, то

$$M(n)\leqslant M(3\cdot 2^{k-1})<3^{k-1}(M(3)+22)\leqslant 3^{k-1}\cdot 35\leqslant\frac{35}{3}n^{\log_2 3}.$$

29. Проверьте, что обычный способ умножения многочленов даёт оценку $M(n)\leqslant n^2+(n-1)^2$.

§ 19. БЫСТРОЕ УМНОЖЕНИЕ ЧИСЕЛ

Перейдём теперь к умножению чисел.

Обозначим через $M(n)$ наименьшее количество операций сложения, вычитания и умножения, выполняемых над числами, меньшими a, требующихся для перемножения двух n-значных чисел, записанных в позиционной системе счисления с основанием a.

Метод умножения почти такой же, как и для многочленов. Для примера укажем, как сделать необходимые изменения в рассуждениях из предыдущего параграфа.

Справедливы неравенства

$$M(2n)\leqslant 3M(n)+19n,\quad M(2n+1)\leqslant 2M(n+1)+M(n)+17n+10.$$

Действительно, применим тождество

$$(f_1 b^{\lceil n/2\rceil}+f_0)(g_1 b^{\lceil n/2\rceil}+g_0)=$$
$$=f_1 g_1 b^{2\lceil n/2\rceil}+(f_1 g_1+f_0 g_0-(f_1-f_0)(g_1-g_0))b^{\lceil n/2\rceil}+f_0 g_0,$$

где числа f_1 и g_1 — $\lfloor n/2\rfloor$-разрядные, а числа f_0 и g_0 — $\lceil n/2\rceil$-разрядные и заметим, что для вычисления произведений $f_1 g_1$ и $f_0 g_0$ требуется $M(\lceil n/2\rceil)+M(\lfloor n/2\rfloor)$ операций, для вычисления разностей f_0-f_1, g_0-g_1 и суммы $f_1 g_1+f_0 g_0$ требуется не более

$$n(1+\lfloor n/2\rfloor-\lceil n/2\rceil)+2(\lfloor n/2\rfloor+\lceil n/2\rceil-1)+2(\lfloor n/2\rfloor+\lceil n/2\rceil)-1=$$
$$=4n-3+n(1+\lfloor n/2\rfloor-\lceil n/2\rceil)$$

операций, так как числа $f_1 g_1$ и $f_0 g_0$ имеют не более чем $2\lfloor n/2\rfloor$ и $2\lceil n/2\rceil$ разрядов соответственно, а в случае чётного n нужно ещё $2\lfloor n/2\rfloor=n$ операций для предварительного сравнения чисел (чтобы не вычитать из меньшего большее). Заметим далее, что для вычисления произведения $(f_1-f_0)(g_1-g_0)$ требуется не более $M(\lceil n/2\rceil)+1$ операций (одна операция для вычисления знака у произведения), для вычисления разности $f_1 g_1+f_0 g_0-(f_1-f_0)(g_1-g_0)=f_1 g_0+f_0 g_1$ требуется не более $2\lceil n/2\rceil+1+2\lceil n/2\rceil-1=4\lceil n/2\rceil$ операций, сложение чисел $f_0 g_0$ и $f_1 g_1 b^{2\lceil n/2\rceil}$ осуществляется «бесплатно» (записи этих чисел просто объединяются в одну запись), а для сложения чисел $f_1 g_1 b^{2\lceil n/2\rceil}+f_0 g_0$ и $(f_1 g_0+f_0 g_1)b^{\lceil n/2\rceil}$ требуется не более $2n-\lceil n/2\rceil+n+1-1=2n+\lfloor n/2\rfloor$ операций (так как число $f_1 g_0+f_0 g_1$ имеет не более $n+1$ разрядов, а младшие $\lfloor n/2\rfloor$ разрядов числа $f_0 g_0$ не участвуют в операциях). В итоге требуется дополнительно $4n-3+n(1+\lfloor n/2\rfloor-\lceil n/2\rceil)+1+4\lceil n/2\rceil+2n+\lfloor n/2\rfloor=7n+3\lceil n/2\rceil+$ $+n(1+\lfloor n/2\rfloor-\lceil n/2\rceil)-2$ операций.

Остальные детали предоставляем додумать читателю.

Обозначим через $Q(n)$ сложность возведения n-разрядного числа в квадрат и такое же обозначение будем использовать для сложности возведения в квадрат многочлена степени $n-1$.

30. Используя тождество $ab=\dfrac{(a+b)^2-(a-b)^2}{4}$, докажите для случая операций с числами неравенство $M(n)\leqslant 2Q(n)+13n+$ $+O(1)$, а для случая операций с многочленами — неравенство $M(n)\leqslant 2Q(n)+6n+4$.

ПРИЛОЖЕНИЕ
ЧТО МОЖНО ВЫЧИСЛИТЬ НА СЧЁТАХ!

Воспользовавшись такой простейшей моделью вычислений, как абак, в России называвшийся просто счётами, можно построить всё здание современной теории алгоритмов. Разумеется, это понятие надо немного идеализировать и придать ему, например, такой вид.

Пусть нам нужно вычислить данную числовую функцию $f(x_1, \ldots, x_n)$. Представим себе, что у нас есть счёты, содержащие n «входных» спиц, на i-й из которых имеется в начальный момент x_i костяшек, одну «выходную» (первоначально пустую) спицу, на которой будет получен результат, и некоторое количество (первоначально пустых) «рабочих» спиц. Каждая спица состоит на самом деле из двух половин, и пока речь шла только о левых половинах. В правой половине каждой спицы помещается потенциально не ограниченный запас костяшек, и по нашему желанию мы можем сделать в любой момент одну из двух операций: либо передвинуть самую левую костяшку из правой половины спицы в «рабочую» левую половину, увеличив тем самым «записанное» в ней число на единицу, либо передвинуть крайнюю правую костяшку из «рабочей» левой половины спицы в правую «запасную» половину, уменьшив тем самым «записанное» в левой половине число на единицу.

Если перейти к терминологии языков программирования, то мы здесь описали систему регистров машины, и две операции, применимые к ним — прибавление единицы и вычитание единицы.

Сама программа вычисления на счётах (или на соответствующей идеализированной машине с неограниченными регистрами) представляет из себя диаграмму, состоящую из кружочков, в которых написаны номера спиц (регистров), после которых стоят знаки плюс или минус, указывающие на операции, которые мы выполняем на этих спицах (регистрах). Из кружочков со знаком плюс выходит одна стрелка, ведущая в какой-то другой кружочек (она указывает какую следующую операцию делать). Из кружочков со знаком минус выходит две такие стрелки. Одна из них помечается специальным значком $*$ и используется только тогда, когда на «рабочей» половине спицы не осталось костяшек (в регистре записан ноль). Тогда операция вычитания единицы, естественно, не может быть выполнена, и просто делается переход к новой вершине диаграммы по указанной стрелке. Если же на спице оставались костяшки (регистр не равен 0), то операция вычитания единицы выполняется и тоже делается переход к новой вершине диаграммы, но, естественно, по второй стрелке. Отметим ещё, что совсем не обязательно, чтобы разные вершины диаграммы выполняли операции с разными спицами.

Теперь, чтобы такая диаграмма могла определить работающую программу, осталось выделить в ней две стрелки — начало и конец работы программы. Первая из них выделяется среди других стрелок тем, что имеет конец в одной из вершин диаграммы, но не имеет начала в вершинах диаграммы, а начинается в специальном кружке со словом «НАЧАЛО», а вторая, наоборот, начинается в одной

из вершин, но не ведёт ни в одну из вершин диаграммы, а ведёт в кружок со словом «КОНЕЦ».

Программа начинает работать со слова «НАЧАЛО» и заканчивает, когда придёт в слово «КОНЕЦ» (но может и «зациклиться» и никогда не закончить работу). Результатом работы программы можно считать число, записанное на «выходном» регистре. Если это число всегда совпадает со значением рассматриваемой функции $f(x_1, \ldots, x_n)$, в случае, если она определена при заданных значениях переменных, и если программа всегда «зацикливается» в случае, если эта функция не определена при заданных значениях переменных, то говорят, что программа (вычисления на счётах!) правильно вычисляет заданную функцию.

С целью сокращения диаграмм у сложных программ можно вместо некоторых вершин, имеющих одну выходную стрелку, использовать не оператор прибавления единицы, а кружок с символическим обозначением какой угодно программы (называемой в этом случае, естественно, подпрограммой).

Работу любой такой программы можно промоделировать и на машине Тьюринга, если изображать состояние абака в каждый момент времени на ленте машины в виде массивов палочек, разделённых пробелами. В возможность обратного моделирования поверить труднее, тем не менее справедливо следующее утверждение, приводимое без доказательства.

Класс числовых функций, вычислимых на абаке, совпадает с классом функций, вычислимых по Тьюрингу.

А как известно, на машине Тьюринга можно промоделировать любые компьютерные вычисления. Значит, и на счётах тоже можно!

31. Приведите пример «зацикливающейся» программы для абака.

32. Приведите пример программы, складывающей содержимое двух регистров и помещающей результат во второй регистр одновременно с обнулением первого регистра.

33. Приведите пример программы, складывающей содержимое двух регистров и помещающей результат во второй регистр, но не изменяющей первый регистр (используйте вспомогательный «рабочий» регистр).

34. Приведите пример программы, перемножающей содержимое двух регистров и помещающей результат в третий регистр одновременно с обнулением первого регистра. (Используйте предыдущую программу в качестве подпрограммы.)

35. Приведите пример программы, перемножающей содержимое двух регистров и помещающей результат во второй регистр без изменения первого регистра. (Используйте вспомогательный регистр и предыдущую программу в качестве подпрограммы.)

36. Покажите, как возводить в степень на счётах. (Используйте предыдущую программу в качестве подпрограммы.)

ЛИТЕРАТУРА

Литература по теме книжки огромна, и приводимый далее список не претендует на полноту. В него включены некоторые источники, использованные автором при подготовки книжки, а также расширяющие и дополняющие её. Они выбраны из числа наиболее доступных, в том числе и по времени издания.

[1] Алфутова Н. Б., Устинов А. В. Алгебра и теория чисел: Сборник задач. — М.: МЦНМО, 2002.

[2] Андреева Е., Фалина И. Системы счисления и компьютерные арифметика. — М.: Лаборатория базовых знаний, 1999.

[3] Булос Д., Джефри Р. Вычислимость и логика. — М.: Мир, 1994.

[4] Васильев Н. Б., Егоров А. А. Задачи всесоюзных математических олимпиад. — М.: Наука, 1988. — (Библиотека математического кружка. Вып. 18).

[5] Воробьёв Н. Н. Признаки делимости. — М.: Наука, 1988. — (Популярные лекции по математике. Вып. 39).

[6] Гальперин Г. А., Толпыго А. К. Московские математические олимпиады. — М.: Просвещение, 1986.

[7] Гарднер М. Математические головоломки и развлечения / Пер. с англ. Ю. А. Данилова / Под ред. Я. А. Смородинского. — М.: Мир, 1999. — (Математическая мозаика. Вып. 1).

[8] Гарднер М. Математические досуги / Пер. с англ. Ю. А. Данилова / Под ред. Я. А. Смородинского. — М.: Мир, 2000. — (Математическая мозаика. Вып. 2).

[9] Гарднер М. Математические новеллы / Пер. с англ. Ю. А. Данилова / Под ред. Я. А. Смородинского. — М.: Мир, 2000. — (Математическая мозаика. Вып. 3).

[10] Гашков С. Б., Чубариков В. Н. Арифметика, алгоритмы, сложность вычислений. — М.: Высшая школа, 2000.

[11] Дадаев Ю. Г. Теория арифметических кодов. — М.: Радио и связь, 1981.

[12] Еленьский Щ. По следам Пифагора. — М.: Детгиз, 1961.

[13] Кнут Д. Искусство программирования: Пер. с англ. — Т. 2. — М.: Вильямс, 2000.

[14] Петцольд Ч. Код. — М.: Русская редакция Майкрософт Пресс, 2001.

[15] Севидж Д. Сложность вычислений. — М.: Факториал, 1998.

[16] Стахов А. П. Коды золотой пропорции. — М.: Радио о связь, 1984.

[17] Фомин С. В. Системы счисления. — М.: Наука, 1980. — (Популярные лекции по математике. Вып. 40).

[18] Штейнгауз Г. Математический калейдоскоп: Пер. с польск. — М.: Наука, 1981. — (Библиотечка «Квант». Вып. 8).

ВЫПУСК 1
В. М. Тихомиров. Великие математики прошлого и их великие теоремы.

ВЫПУСК 2
А. А. Болибрух. Проблемы Гильберта (100 лет спустя).

ВЫПУСК 3
Д. В. Аносов. Взгляд на математику и нечто из неё.

ВЫПУСК 4
В. В. Прасолов. Точки Брокара и изогональное сопряжение.

ВЫПУСК 5
Н. П. Долбилин. Жемчужины теории многогранников.

ВЫПУСК 6
А. Б. Сосинский. Мыльные плёнки и случайные блуждания.

ВЫПУСК 7
И. М. Парамонова. Симметрия в математике.

ВЫПУСК 8
В. В. Острик, М. А. Цфасман. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые.

ВЫПУСК 9
Б. П. Гейдман. Площади многоугольников.

ВЫПУСК 10
А. Б. Сосинский. Узлы и косы.

ВЫПУСК 11
Э. Б. Винберг. Симметрия многочленов.

ВЫПУСК 12
В. Г. Сурдин. Динамика звёздных систем.

ВЫПУСК 13
В. О. Бугаенко. Уравнения Пелля.

ВЫПУСК 14
В. И. Арнольд. Цепные дроби.

ВЫПУСК 15
В. М. Тихомиров. Дифференциальное исчисление (теория и приложения).

ВЫПУСК 16
В. А. Скворцов. Примеры метрических пространств.

ВЫПУСК 17
В. Г. Сурдин. Пятая сила.

ВЫПУСК 18
А. В. Жуков. О числе π.

ВЫПУСК 19
А. Г. Мякишев. Элементы геометрии треугольника.

ВЫПУСК 20
И. В. Ященко. Парадоксы теории множеств.

ВЫПУСК 21
И. Х. Сабитов. Объёмы многогранников.

ВЫПУСК 22
А. Л. Семёнов. Математика текстов.

ВЫПУСК 23
М. А. Шубин. Математический анализ для решения физических задач.

ВЫПУСК 24
А. И. Дьяченко. Магнитные полюса Земли.

ВЫПУСК 25
С. М. Гусейн-Заде. Разборчивая невеста.

ВЫПУСК 26
К. П. Кохась. Ладейные числа и многочлены.

ВЫПУСК 27
С. Г. Смирнов. Прогулки по замкнутым поверхностям.

ВЫПУСК 28
А. М. Райгородский. Хроматические числа.

ВЫПУСК 29
С. Б. Гашков. Системы счисления и их применение.