Информатика

# Новости кубологии

*Кандидат физико-математических наук*

*В. ДУБРОВСКИЙ,*
*А. КАЛИНИН*

Лучшая головоломка XX века, придуманная 13 лет назад венгерским преподавателем дизайна Эрне Рубиком, упорно сопротивляется попыткам разгадать все ее тайны. Сразу после изобретения кубика Рубика начался поиск самого короткого пути к ее решению. Первые разработанные алгоритмы требовали 200—300 ходов (поворотов граней) для того, чтобы вернуть кубик в исходное состояние.

Постепенно длина алгоритмов (т. е. минимальное число ходов, гарантирующее решение) сокращалась. Это происходило за счет изменения последовательности сборки разных частей головоломки (улучшения стратегии) и применения более коротких операций для перестановки и правильной ориентации маленьких кубиков (улучшения тактики). Ставший самым популярным «послойный» алгоритм кубика Рубика («Квант», 1983, № 9) осуществляет сборку не более чем за 108 ходов). Совершенствуя его, удается уменьшить это число до 86 ходов. Дальнейшие улучшения требуют резкого увеличения количества формул, которые необходимо держать в голове или на бумаге в процессе сборки. В период всеобщего увлечения кубиком (1981—1983 гг.) редакция «Кванта» получила не менее десятка объемистых разработок, авторы которых, проделав гигантскую работу по поиску новых операций, получили алгоритмы, позволяющие решать «головоломку века» в среднем за 50—60 ходов*). Наиболее основательную работу проделал московский инженер А. Карасев, разработавший таблицы для сборки кубика из 5412 операций, разбитых на 22 группы по 246 формул в каждой группе!

Одновременно с любителями решать головоломку, держа ее в руках, неприступный кубик штурмовали и программисты. Сначала успеха добились менее «безумные» из них — те, кто взялись за Малый кубик размером 2Х2Х2 кубичка. Задачу они решали с конца: исходя из правильного состояния кубика, программа начинает «разрушать» его, получая и запоминая результаты всевозможных поворотов граней. Если какая-либо расцветка кубика появляется повторно, то соответствующая операция игнорируется, так что в памяти компьютера остаются только самые короткие формулы. В результате был получен список всех возможных состояний Малого кубика с указанием, после каких поворотов граней они впервые появились. Этот список никогда не был ни опубликован, ни напечатан хотя бы в одном экземпляре. Причина не столько в его громадных размерах, сколько в том, что из-за таких размеров слишком трудно найти в списке нужную вам в данный момент операцию.

Если подсчитать количество состояний Малого кубика после определенного числа поворотов граней, то получится следующая таблица:

1 расцветка — после 0 поворотов граней,
9 расцветок — после 1 поворота граней,
54 расцветки — после 2 поворотов граней,
321 расцветка — после 3 поворотов граней,
1 847 расцветок — после 4 поворотов граней,
9 992 расцветок — после 5 поворотов граней,
50 136 расцветок — после 6 поворотов граней,
227 536 расцветок — после 7 поворотов граней,
870 072 расцветки — после 8 поворотов граней,

*) Подчеркнем, что здесь речь идет именно о среднем числе ходов, а точнее, о числе ходов, которое обычно требовалось самим авторам алгоритмов для сборки кубика их методом.

**1 887 748 расцветок — после 9 поворотов граней,**
**623 800 расцветок — после 9 поворотов граней,**
**2 664 расцветки — после 11 поворотов граней.**

Поворачивая грани 12 раз, компьютер не нашел ни одного нового состояния Малого кубика. Следовательно, чтобы решить головоломку из 8 маленьких кубичков, всегда достаточно сделать не более 11 ходов. Чаще всего хватит и 10 ходов.

Малый кубик есть не что иное, как 8 угловых кубичков классического кубика Рубика. Но в последнем — 26 кубичков, а это уже колоссально усложняет задачу перебора. Кубик Рубика может иметь $N = 4{,}3 \times 10^{19}$ различных расцветок. Если ваша программа будет тратить всего 1 микросекунду на получение и анализ одной расцветки, то понадобится 1,4 миллиона лет, чтобы закончить всю работу. Вот почему в начале статьи мы назвали «безумными» тех программистов, что взялись за решение этой задачи.

Из этих программистов первого впечатляющего успеха добился английский математик М. Тистлетуэйт, который разработал совершенно новый алгоритм, позволявший всегда упорядочить кубик Рубика не более чем за 52 хода («Математика волшебного кубика», «Квант», 1982, № 8). Хотя в принципе с помощью этого алгоритма можно собрать кубик и вручную, реально его может выполнить только компьютер. В дальнейшем этот алгоритм удалось несколько улучшить — сначала этого добился сам англичанин, а пару лет назад X. Клоостерман из Голландии довел длину алгоритма до 42 ходов. Важно отметить, что эта граница обоснована строго (а не эмпирически), т. е. доказано, что из любого состояния кубика Рубика можно вернуться в правильное не более чем за 42 хода, причем данный рекордный алгоритм не может гарантировать лучшего результата. (Это не означает, что другой алгоритм не может оказаться короче.) Конечно, это доказательство существенно использует компьютер (как, например, и относительно недавнее решение знаменитой «проблемы четырех красок»): для каждого из этапов сборки был осуществлен полный перебор вариантов, число ходов, понадобившееся в худшем случае, и принимается за «длину» данного этапа.

Совсем недавно нового достижения добился немецкий математик Герберт Коцемба. Он был среди тех «менее безумных», кто 10 лет назад победили Малый кубик, но затем примкнул к «самым самым» и, возможно, уже остался единственным, кто до наших дней продолжал штурм головоломки века. Теперь к нему пришел заслуженный успех. Он разработал алгоритм и написал программу, которая решает головоломку Рубика менее чем за 21 ход! Сразу скажем, что эта оценка длины, в отличие от предыдущей, эмпирическая: все состояния кубика, которые предлагались программе Коцембы, были упорядочены не более чем за 21 ход. В частности, программа нашла более короткие решения для многих задач на составление узоров на кубике (или пасьянсов), весьма популярных в «золотую эру» кубика. Нет никакой гарантии, однако, что состояний, требующих больше 21 поворота по программе Коцембы, не существует вовсе. Более того, такую гарантию мог бы дать только полный перебор вариантов для каждого этапа программы (их два), а он пока не под силу даже программе Коцембы и его более мощному, чем у предшественников, компьютеру.

## Идея алгоритма Герберта Коцембы

Собственно говоря, то, что придумал Коцемба, не является «алгоритмом сборки кубика» в том смысле, как его обычно понимают в «кубологии».

Обычные алгоритмы сборки представляют собой наборы правил, позволяющих для любого заданного состояния кубика поставить некую ближайшую цель и достичь ее, выполнив последовательность поворо-

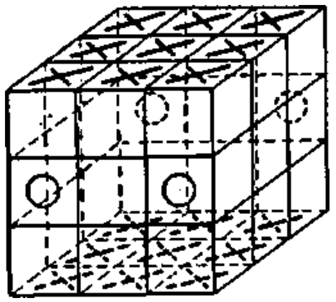

*Рис. I. Промежуточное состояние кубика Рубика.*

тов граней, предписываемую правилами в данной ситуации. Тем самым кубик переводится в новое состояние, более близкое к правильному (по крайней мере, с точки зрения данного алгоритма). Например, цель может состоять в том, чтобы найти неправильно стоящий угловой кубичек и переместить его в свой угол, не трогая остальные угловые кубички. И таких маленьких шажков приходится делать очень много.

В алгоритме Коцембы ставится только одна промежуточная цель — кубик надо перевести в одно из состояний, которые так и названы — промежуточными. Они характеризуются тем, что любое промежуточное состояние можно получить из правильного (а значит, и наоборот — превратить его в правильное), поворачивая четыре боковые грани только на 180°, а верхнюю и нижнюю — на произвольный угол (естественно, кратный 90°). Более наглядное описание семейства промежуточных состояний дает специальная расцветка кубика, изображенная на рисунке 1: верхняя и нижняя грани красятся одним цветом, а на каждом из реберных кубичков горизонтального среднего слоя один из двух квадратиков красится вторым цветом; остальную поверхность кубика вообще можно не закрашивать. Первая цель (задача первого этапа) алгоритма Коцембы — восстановить такую раскраску из хаотического исходного состояния. При этом, конечно, можно пользоваться любыми поворотами. На втором этапе применяются только повороты, перечисленные выше. Легко понять, что они автоматически сохраняют нашу вспомогательную раскраску. По существу, на втором этапе происходит только установка каждого кубичка на его место. А благодаря сохранению вспомогательной раскраски правильная ориентация кубичков на своих местах будет обеспечена автоматически. Таким образом, число промежуточных состояний равно числу допустимых перестановок кубичков (т. е. перестановок, получаемых из правильной поворотами граней), при которых реберные кубички среднего слоя остаются в этом слое. Его нетрудно вычислить: четыре кубичка среднего слоя переставляются 4! способами, остальные реберные кубички — 8! способами, угловые — тоже 8! способами. Общее число всевозможных перестановок — $(8!)^2$X X4! — нужно еще поделить пополам, чтобы выделить допустимые перестановки (см. статью «Математика волшебного кубика»). Итак, получается $N_2=(8!)^2 * 4!/2=19,5*10^9$ промежуточных состояний. Эта величина дает представление о порядке числа вариантов, которое должна держать в своей памяти машина, подыскивая достаточно короткую операцию для упорядочивания промежуточного состояния

На первом же этапе программу интересует не полное описание заданного состояния кубика (т. е. не все $4,3*10^{19}$ возможных вариантов), а только исходное расположение цветов вспомогательной раскраски, которое нужно привести к стандартному виду (рис. 1). Другими словами, здесь важны ориентации всех кубичков и места, занимаемые четырьмя кубичками среднего слоя. Таким образом, число $N_1$ вариантов, рассматриваемых на первом этапе, равно

(число допустимых ориентации 8 угловых кубичков) X (число допустимых ориентаций 12 реберных кубичков) X (число расположений кубичков горизонтального слоя)$=3^7*2^{11}*C^4_{12}=2,217*10^9$

(почему $3^7$ и $2^{11}$, а не $3^8$ и $2^{12}$, объясняется в упомянутой выше «Математике волшебного кубика»).

Естественно, $N_1 N_2$ — это общее число допустимых состояний кубика, и оно на 9 порядков больше числа $N_1+N_2$ вариантов, охватываемых программой. Тем не менее, даже это уменьшенное число все еще слишком велико, чтобы можно было произвести исчерпывающий перебор случаев и запомнить его результаты. Поэтому машина не знает заранее, какое решение она выдаст для введенного в нее исходного состояния кубика. Она ищет решение «в присутствии заказчика» и выдает, возможно, не крат чайший, но достаточно короткий вариант.

Отметим, что метод Коцембы также восходит к алгоритму Тистлетуэйта. Однако в последнем используются не один, а три вложенных друг в друга класса промежуточных состояний, отвечающих поэтапному сужению набора разрешенных поворотов; второй из этих трех классов и составляют промежуточные состояния Коцембы. Понятно, что если вам нужно добраться из пункта *A* (исходное состояние) в пункт *B* (правильное состояние) и по дороге обязательно посетить пункт C (промежуточное состояние), то такой путь *ACB* может оказаться длиннее прямого пути *A B,* даже если его отрезки *AC* и *CB* проходятся оптимально. А если еще и на пути из А в С надо завернуть в *D,* а из *С в В* — в дополнительное промежуточное состояние *E,* то получится еще более длинный путь *ADCEB.* Зато каждый из отрезков этого пути уже настолько сокращается, что полный перебор случаев становится возможным. Так и появились рекордные результаты Тистлетуэйта и Клоостермана.

### Как работает программа Коцембы

Грубо говоря, Коцемба заставляет машину просматривать всевозможные цепочки поворотов, разрешенных на соответствующем этапе, и ловить момент, когда цель этого этапа будет достигнута. Однако при таком лобовом подходе объем просматриваемых вариантов был бы слишком велик. Так, первый ход можно сделать 6Х Х3=18 способами (любую из шести граней можно повернуть на один из трех углов — 180°, ±90°), на втором и каждом из следующих ходов число способов равно 15, так как нет смысла дважды подряд поворачивать одну грань. Таким образом, возникает «дерево вариантов», от каждой ветви которого отходят пять следующих ветвей. (На самом деле, начиная с определенного момента некоторые ветви будут срастаться, потому что разные цепочки ходов могут порождать одинаковые преобразования кубика.) Число цепочек ходов длины, не превосходящей *n*, равно сумме геометрической прогрессии $18(1+15+\ldots+15^{n-1})$.

Между прочим, только при *n*=18 это число превысит общее число *N* состояний кубика, а значит, заведомо найдутся состояния, которые нельзя упорядочить меньше чем за 18 ходов. В действительности, из-за сращивания ветвей дерева вариантов число 18 можно еще увеличить. (В свое время промелькнуло сообщение о том, что доказано существование состояний, не решаемых быстрее чем за 21 ход; впрочем, оно могло быть не вполне достоверным ) Как видим, результаты, показываемые программой Коцембы, близки к наилучшим.

Сокращения перебора Герберт Коцемба добился с помощью специальной фильтрующей программы. Она хранит определенную информацию о всех цепочках из, скажем, не более чём 8 ходов, и позволяет отсеивать состояния, которые заведомо не упорядочиваются (в смысле 1-го или 2-го этапов алгоритма) такими цепочками. Начав «сборку», компьютер настраивается выполнить первый этап за 10 ходов. Он порождает первые два хода и включает фильтр на 8 ходов; если возникшее состояние не отсеется, производится третий ход и включается фильтр на 7 ходов, и т. д. Если на каком-то шагу произойдет отсев, надо поменять предыдущий сделанный ход. Пока что для всех позиций, предлагавшихся программе,

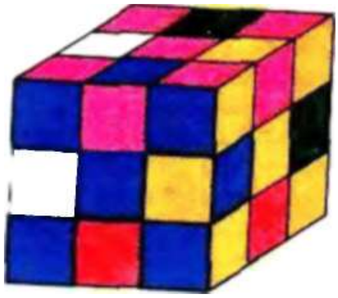

*Рис. 2. Пасьянс «реверс» — осе реберные кубички перевернуты на своих местах.*

удавалось осуществить первый этап не более чем за 10, а второй — не более чем за 14 ходов.

Для реализации своего алгоритма Коцемба использовал персональный компьютер Atari ST с памятью 1 Мбайт и частотой 8 Мгц. Ядро программы — генератор операций с блоком проверки, написанные на Ассемблере. Размеры этого ядра — менее 500 байт! Остальные части программы, которые не оказывают существенного влияния на быстродействие, выполнены на Бейсике. В результате среднее время нахождения одного решения — 1,5 минуты (10 секунд на первый этап и 85 секунд на второй). Работа построена так: в компьютер вводят описание конкретного кубика (расположение и ориентацию всех кубичков). После этого машина ищет и запоминает лучшие (не более 13 ходов) пути для достижения промежуточного состояния, а затем для самого короткого из них включается второй этап. Если суммарная длина полученной после этого операции больше 21 хода, компьютер возвращается к этапу 1 и пытается найти более короткое решение на базе другого промежуточного состояния, возможно, увеличив длину первого этапа. Предъявляя компьютеру различные варианты запутанного кубика, Герберту всегда удавалось обнаружить операцию не длиннее 21 хода, хотя бы на это и уходило много часов работы программы.

Герберт Коцемба обратился ко всем любителям кубика с предложением присылать ему варианты запутанного кубика, к которым не удается вручную найти операции короче 21 хода. А мы в 1982—83 гг. проводили конкурс среди читателей популярного еженедельника «Собеседник» по лучшим узорам (пасьянсам) на гранях кубика Рубика. Тогда победители нашли операции короче 21 хода ко всем узорам, кроме одного (см. рис. 2), и теперь мы послали эту задачу в Дармштадт, где живет автор рекордного алгоритма.