Занимательный компьютер

Множество Мандельброта и родственные ему множества Жюлиа

А. К. ДЬЮДНИ

С МОМЕНТА публикации первой статьи о множестве Мандельброта в октябрьском номере нашего журнала за 1985 г. оно приобрело самую большую популярность у любителей математических трюков. Их привлекает как внешняя красота множества, так и его глубокая сущность. На самом деле красота множества Мандельброта — это лишь вуаль, за которой скрывается его значимость: в замысловатом узоре из тонких линий и завитков вблизи границы множества закодированы различные формы хаоса и упорядоченности (см. верхний рисунок на с. 90).

В упомянутой статье наш рассказ о множестве Мандельброта был далеко не полным. Множество имеет некоторую важную связь со свойствами устойчивости и хаоса в динамических системах. Эта связь проходит через родственные множества, называемые множествами Жюлиа по имени французского математика Гастона Жюлиа. Каждой точке, принадлежащей множеству Мандельброта (и не принадлежащей ему), соответствует одно множество Жюлиа. Множества Жюлиа с их фрактальной природой тоже по-своему красивы (см. нижний рисунок на с. 90). Но прежде чем рассказывать о них, мы вернемся еще раз к множеству, носящему имя Бенуа Мандельброта, ученого, работающего в Исследовательском центре Томаса Уотсона фирмы IBM в Йорктаун-Хайтсе (шт. Нью-Йорк).

Множество Мандельброта — обитатель комплексной плоскости, т. е. обычной плоскости, каждая точка которой характеризуется двумя координатными значениями. Точнее говоря, каждая точка комплексной плоскости представляется числом вида $a + bi$. Числа a и b можно считать координатами точки: a — вещественная часть комплексного числа $a + bi$, а b — его мнимая часть. Число i отчасти служит для того, чтобы различать эти две координаты между собой. Комплексные числа можно складывать, суммируя их соответственные координаты по отдельности; результат представляет собой другое комплексное число. Их можно также умножать друг на друга, следуя при этом правилам умножения многочленов:

$$\begin{array}{r@{}l} & 3 + 7i \\ \times & 2 - 4i \\ \hline & 6 + 14i \\ & \quad - 12i - 28i^2 \\ \hline & 6 + 2i - 28i^2 \end{array}$$

Чтобы представить результат в виде комплексного числа, член $28i^2$ нужно привести к стандартной форме, воспользовавшись самым важным свойством мнимых величин, а именно тем, что $i^2 = -1$. Таким образом, выражение $6 + 2i - 28i^2$ приводится к виду $34 + 2i$. Теперь мы можем представить основную формулу, которая открывает нам множество Мандельброта, порождает также множества Жюлиа и в некотором смысле превращает порядок в хаос:

$$z \leftarrow z^2 + c.$$

Здесь z и c — это комплексные числа, каждое из них имеет мнимую и вещественную части. Число z возводится в квадрат и c прибавляется к результату согласно правилам умножения и сложения комплексных чисел. Формула как бы оживает, если начать последовательно вычислять значения сумм, подставляя в формулу каждый раз значение z, полученное на предыдущем шаге. Получающаяся в результате этого итерационного процесса последовательность комплексных чисел образует причудливый узор на комплексной плоскости. На каждом шаге новое комплексное число z лежит на некотором расстоянии от предшествующего. Это расстояние очень важно для вычисления множества Мандельброта.

Интересно представить полученную последовательность комплексных чисел (точек на комплексной плоскости) как блуждания исходной точки. Стремится ли она к бесконечности, все дальше удаляясь от начала координат комплексной плоскости? Некоторые комплексные числа действительно стремятся в бесконечность, другие навсегда ограничены в своем движении определенной областью, имеющей сложную форму. Их тюрьма, или область, где они содержатся, имеет фрактальные стены.

Как уже говорилось, итерационный процесс длится до бесконечности. Однако, каким образом выбираются константа c и исходное значение z? Можно, например, сделать число z равным нулю и выбирать различные значения c. Вырвется ли узник в данном случае на свободу? Будем повторять эксперимент снова и снова, систематически варьируя значение c в определенной области комплексной плоскости. Если узнику удается бежать, окрашиваем точку c в белый цвет, в противном случае делаем ее черной. Стены области, из которой не удается бежать, принимают форму множества Мандельброта. Если вместо того, чтобы окрашивать всех беглецов в белый цвет, придавать им окраску в зависимости от скорости побега, то изображение становится еще красивее.

Согласно только что описанному правилу, исходное значение числа z равно 0, или точнее, $0 + 0i$. А что изменится, если процесс начать с какого-нибудь другого значения, скажем, $z = 3{,}5 + 6i$? Примет ли результирующее множество другую форму? Действительно, в результате мы всегда получаем деформированную версию множества Мандельброта. Обычно предпочтение отдается его канонической форме.

Если следовать противоположному правилу, когда значение c фиксировано, а z играет роль исходной точки, получающееся в результате итерационного процесса множество уже отличается по виду от множества Мандельброта. Оно, или вернее его граница, называется множеством Жюлиа. Множество Жюлиа не единственно, на самом деле имеется целое множество таких множеств: для каждого фиксированного значения c в формуле итераций мы получаем свое, отличное от других, множество Жюлиа, заполненное «узниками».

Я был вдохновлен на то, чтобы вернуться еще раз к множеству Мандельброта, прочтя книгу сотрудников Бременского университета Х.-О. Пайтгена и П. Рихтера «Красоты фракталов» (см. библиографию к статье на с. 104). Эта крупноформатная книга, содержащая потрясающие черно-белые и цветные иллюстрации, с одинаковым успехом может служить и математическим учебным пособием, и украшением журнального столика.

Вся теория, касающаяся множества Мандельброта и связанных с ним множеств Жюлиа, а также других аналогичных сложных систем, изложена в книге в виде математических теорем с доходчивым их описанием.

Рассмотрим смысл одной из теорем. Читатель, написавший программу для построения изображений множеств Жюлиа, возможно заметит, что для некоторых значений c множества получаются связными и графически выглядят как единое целое, в то время как для других значений c они оказываются несвязными. Что же обусловливает это различие? Ответ и прост и изящен: если выбранная точка c принадлежит множеству Мандельброта, то соответствующее множество Жюлиа будет связным. Если же c не принадлежит множеству Мандельброта, то множество Жюлиа будет несвязным.

Можно сделать нечто вроде мультипликационного фильма, иллюстрирующего смысл данной теоремы. Проведем прямую L от произвольной точки, принадлежащей множеству Мандельброта, к другой точке, не принадлежащей ему, и представим себе, что точка c медленно, но неуклонно продвигается вдоль линии L из внутренней области множества Мандельброта по направлению к его границе. Соответствующее множество Жюлиа будет становиться все более сжатым, постоянно уменьшаясь в размере, а когда точка c достигнет границы множества Мандельброта, множество Жюлиа сожмется до хрупкого дендритного скелета, практически не занимающего никакой площади. Когда c выйдет за границу множества Мандельброта, сопутствующее ему множество Жюлиа как бы взрывается, превращаясь во фрактальную пыль.

Читатели, которые пожелают и смогут написать программу, могут исследовать свойства множества Мандельброта и сопутствующих ему множеств Жюлиа, реализовав некоторые основные алгоритмы на том языке, которым они обычно пользуются. Во всех подобных алгоритмах присутствует итерационный процесс, зависящий существенным образом от следующей теоремы: если абсолютная величина итерируемой переменной z достигает 2, то ей суждено убежать в бесконечность, откуда она уже никогда не вернется. В основном именно этот критерий отличает беглецов от узников. Алгоритм содержит 100 шагов итераций, позволяющих переменной z достичь значения 2. Этот критерий, конечно, не срабатывает во всех 100% случаев, поскольку некоторое относительно небольшое количество беглецов не успевает достичь абсолютной величины 2 за 100 итераций. Можно было бы сделать

Множество Мандельброта в виде озера; горы представляют динамику окружающих точек число итераций равным, скажем, 1000, чтобы получить более точную картинку, но для этого потребовалось бы много машинного времени, даже на мощных компьютерах.

Абсолютная величина комплексного числа $a + bi$ равна просто квадратному корню из $a^2 + b^2$, другими словами, она представляет собой расстояние от начала координат, или нулевой точки. Основной алгоритм итераций имеет следующий вид:

n $\leftarrow 0$
while $n < 100$ and abs $(z) < 2$
    $z \leftarrow z^2 + c$
    $n \leftarrow n + 1$
раскрасить данную точку

Здесь n — индекс цикла с исходным значением 0. На каждом шаге цикла while, управляющего итерационным процессом, значение переменной n увеличивается на 1. Цикл while продолжается, «проворачивая мясорубку» итерационной формулы до тех пор, пока n не достигнет 100 или абсолютная величина z не достигнет 2. При выполнении любого из этих условий алгоритм выходит из цикла. Читателю предоставляется самому выбрать способ раскраски точек изображения. Этот способ, конечно, должен каким-то несложным образом зависеть от величины n, т. е. от той скорости, с которой точке удалось достичь расстояния 2, а также от того факта, удалось ли ей достичь этого расстояния. Читателям следует также не упускать из виду, что точка изображения на экране будет иметь, вообще говоря, другие координаты, отличающиеся от координат точки на комплексной плоскости.

Программа, написанная читателями, должна содержать отдельную процеруру вычисления абсолютной величины z, которая в приведенном выше алгоритме была представлена как abs(z). На самом деле, поскольку в большинстве языков программирования отсутствуют средства для работы с комплексными числами, переменную z нужно хранить в виде двух частей, скажем x (вещественная часть) и y (мнимая часть), то же касается и числа c, части которого можно обозначить соответственно как a и b. Следующий алгоритм будет уже несколько ближе к работающей программе:

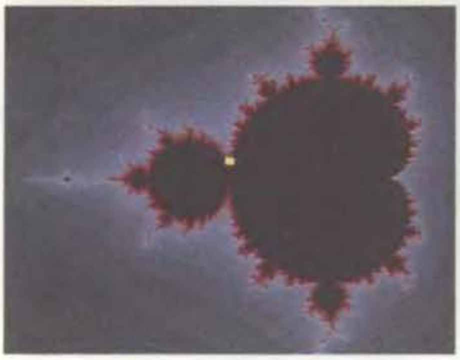
$n \leftarrow 0$
while $n < 100$ and $x^2 + y^2 < 4$
    $xx \leftarrow x^2 + y^2 + a$
    $y \leftarrow 2xy + b$
    $x \leftarrow xx$
    $n \leftarrow n + 1$
раскрасить данную точку

Наблюдательный читатель, наверное, заметит маленький трюк, введенный в эту версию итерационного процесса: вместо того чтобы сравнивать квадратный корень из выражения $x^2 + y^2$ с 2, мы сравниваем само это выражение с 4. Результат тот же, но с помощью этого приема можно избежать многократного обращения к функции извлечения квадратного корня, выполнение которой требует относительно большого времени. Переменная xx временно содержит только что вычисленное значение x, пока вычисляется новое значение y. Таким образом, старое значение x сохраняется для дальнейших вычислений, пока его не заменит значение xx.

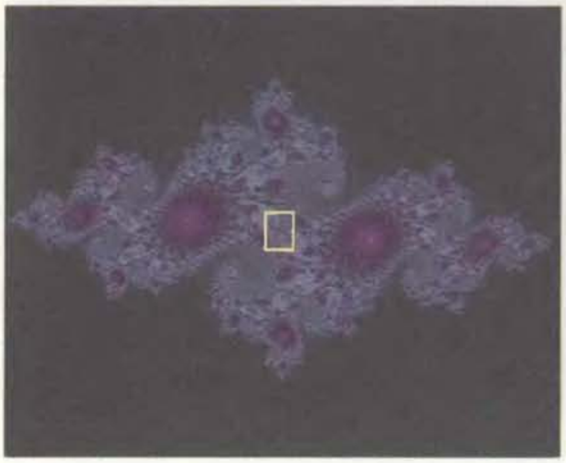
Программа, которую я назвал MANDELZOOM, плод моих первых попыток в данной области, предпринятых еще два года назад, будет представлена теперь в несколько более детальном виде. Она содержит основной итерационный алгоритм внутри цикла, в котором систематически варьируется комплексное число c и, как следствие, его части a и b. Если экран, на котором строится изображение, имеет размер 100×100 пикселов (точечных элементов изображения, образующих квадратную решетку), то программа будет основана на следующем двойном цикле:

gap $\leftarrow$ *side*/100
a $\leftarrow$ *acorner*
for $j \leftarrow 1$ to 100
    $a \leftarrow a +$ *gap*
    $b \leftarrow$ *bcorner*
    for $k \leftarrow 1$ to 100
        $b \leftarrow b +$ *gap*
        $x \leftarrow 0$
        $y \leftarrow 0$
        [*алгоритм итераций*]

Однако, прежде чем программа MANDELZOOM доходит до этих операторов, она позволяет пользователю задать комплексное число, располагающееся в одном из углов квадрата, на котором строится изображение. Этот угол будет иметь координаты *acorner*, *bcorner*, наименьшие значения a и b, принимаемые комплексными числами в пределах данного квадрата. Квадрат, выбираемый пользователем программы, является как бы окном, через которое мы можем наблюдать изображение. Его можно сделать очень маленьким и тогда мы будем наблюдать в увели-




Множество Мандельброта (слева), *часть которого* (в квадрате) *увеличена* (справа)

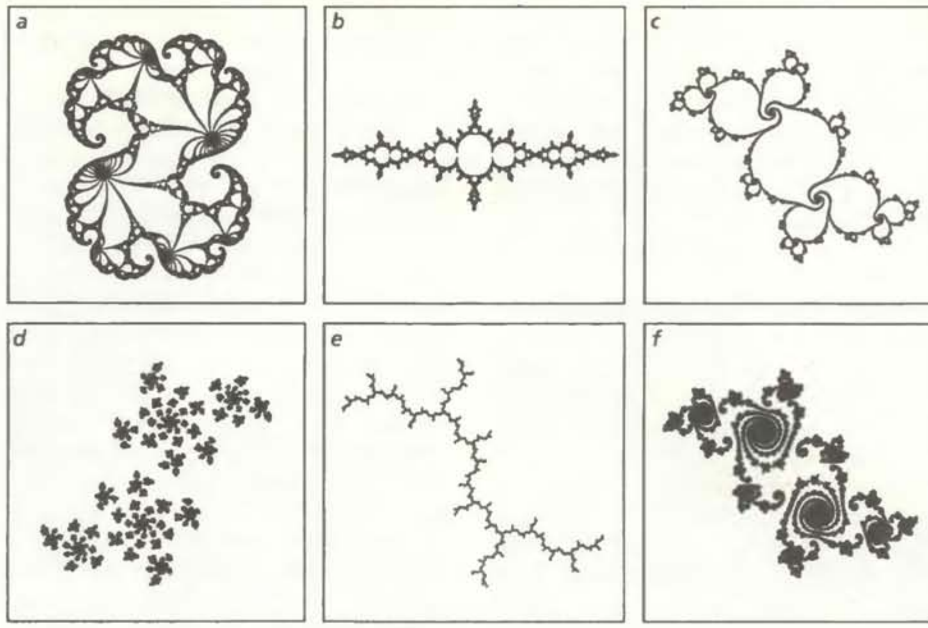



Множество Жюлиа (слева), *соответствующее центральной части верхнего правого рисунка*; *увеличенная часть* (справа) ченном виде ту часть множества, над которой располагается окно. Программа требует также указать значение переменной *side*, т. е. размер стороны квадрата на комплексной плоскости. Затем алгоритм вычисляет расстояние между последовательными комплексными числами *c*, задавая тем самым соответствующие приращения переменным *a* и *b* (это расстояние обозначено как *gap*).

Значения индексов *j* и *k* не участвуют ни в каких вычислениях внутри двойного цикла, поэтому эти индексы можно заменить какими-нибудь величинами, представляющими интерес. Например, вместо того чтобы пробегать значения от 1 до 100, индексы *j* и *k* могли бы пробегать через 100 последовательных точек на координатных осях экрана. После того как основная итерационная процедура присвоила определенное цветовое значение итерируемой точке *z*, точка с координатами (*j*, *k*) окрашивается в соответствующий цвет.

Рассмотрение программы MANDELZOOM будет неполным, если не упомянуть о модификации, предложенной Пайтгеном. Вместо того чтобы сравнивать абсолютную величину итерируемой переменной *z* с 2, можно в качестве порогового значения принять 100 или даже 1000. В конце концов, после того как абсолютная величина комплексной переменной достигла 2, она растет очень быстро и достигает значений порядка 100 или 1000 за несколько шагов итерационного процесса. И все же при различных исходных значениях итерируемые точки проходят пороговые величины с различными скоростями. Скорости тоже можно окрашивать, причем с плавными переходами между различными цветами. Красный цвет может, например, постепенно переходить в оранжевый, конечно, при условии, что цветовая палитра компьютера достаточно богата. Во всяком случае, цветные изображения, приведенные в данной статье, были получены именно этим способом. Пайтген считает, что распределение скоростей напоминает электрическое поле, как бы создаваемое вокруг множества Мандельброта. Градиенты напряженности «поля» представлены в воображаемом ландшафте Мандельброта (см. рисунок на с. 89) в виде склонов горного хребта, окружающего то, что можно было бы назвать озером Мандельброта.

Хотя я не считаю, что слово ZOOM стало для меня навязчивым, я все же должен последовать примеру программы MANDELZOOM и назвать программу, генерирующую изображения множеств Жюлиа, JULIAZOOM.



Шесть множеств Жюлиа; некоторые из них связные (a, b, c и e), *а другие нет* (d и f)

(Одно из значений слова ZOOM — это «рассматривать под большим увеличением».) Здесь мы также можем вплотную приблизиться к множеству и исследовать его как бы под очень сильной лупой. В программе JULIAZOOM применяется тот же основной алгоритм итераций, что и в программе MANDELZOOM, однако здесь у него уже несколько иная, так сказать, оправа.

Сначала программа запрашивает у пользователя значения величины *xcorner*, *ycorner* и *side* («угол *x*», «угол *y*» и «сторона» соответственно). Она также запрашивает значение *c* в виде его составляющих *a* и *b*. Затем выполняется двойной цикл, кое в чем существенно отличающийся от цикла программы MANDELZOOM:

```
gap ← side/100
x ← xcorner
for j ← 1 to 100
    x ← x + gap
    y ← ycorner
    for k = 1 to 100
        y ← y + gap
        [алгоритм итераций]
```

Основной алгоритм итераций окрашивает точки экрана в зависимости от скорости, с которой итерационный процесс достигает (или вообще не достигает) порогового значения 2. Некоторые из наиболее эффектных графических результатов были получены при использовании самых простых правил окрашивания точек. На цветных мониторах даже трех цветов, присваиваемых по следующему правилу, оказывается достаточно, чтобы получить прекрасные изображения: окрасить в первый цвет точки со значениями *n* от 0 до 10, вторым цветом — точки со значениями *n* от 11 до 20, третьим цветом — точки со значениями *n* от 21 до 30, затем для следующих десяти значений *n* вернуться к первому цвету и т. д. На нецветных мониторах можно воспользоваться черно-белыми (или зелено-желтыми) контрастами, чередуя оба цвета для каждых десяти последовательных значений *n*.

Вооружившись работающей версией программы MANDELZOOM или JULIAZOOM (а может быть, обеими программами), читатели смогут заняться своими собственными исследованиями этих красивых и интересных фрактальных множеств. Можно побродить по комплексной плоскости в окрестностях множеств или же рассмотреть какие-то области под сильным увеличением, воспользовавшись компьютерным микроскопом, который был описан выше. Вплоть до предельной разрешающей способности, зависящей от точности арифметических операций у того или иного компьютера, эти множества предстают перед наблюдателем в виде удивительных узоров. Будущим исследователям микрокосма фрактальных объектов можно порекомендовать следующие интересные области, с координатными значениями, изменяющимися в пределах:

Множества Жюлиа:
x и *y* от $-1{,}8$ до $+1{,}8$
Множество Мандельброта:
x от $-2{,}25$ до $+0{,}75$ и
y от $-1{,}8$ до $+1{,}5$

В сентябрьском номере журнала

мы затронули тему хаоса в динамических системах, таких как механические маятники и электронные схемы. Свойства этих систем характеризуются простой итерационной формулой, в качестве переменных здесь выступают не комплексные, а обычные вещественные числа:

$$x \leftarrow rx(1 - x).$$

Эта формула, очевидно, квадратична: если раскрыть скобки в правой части, то появится член, содержащий x в квадрате. В зависимости от выбранного значения параметра r, формула ведет себя либо просто, либо странно, если ее начать итерировать. При каждом значении r итерируемая переменная x образует орбиту, т. е. набор значений, которые x систематически посещает. В районе критического значения параметра 3,5699 переменная начинает быстро и более или менее непредсказуемо осциллировать между некоторыми значениями. Такое поведение формулы соответствует ситуации, когда описываемая ею физическая система, будь то двойной маятник или электронная схема, никак не может найти положение стабильности. Ее состояние постоянно меняется непредсказуемым образом, другими словами, наступает хаос.

Аналогичное явление наблюдается и в поведении комплексной итерационной формулы, рассматриваемой в данной статье, $z \leftarrow z^2 + c$. Однако в этом случае для каждого значения параметра c возникает уже не одна орбита — аттрактор. Их количество зависит от того, каким было выбрано исходное значение переменной z. Если абсолютная величина исходного значения z относительно мала, то переменная будет стремиться к определенной точке. Если же абсолютная величина исходного значения z будет достаточно большой, то в дальнейшем она будет неограниченно возрастать, и аттрактором будет уже бесконечность. И определенная точка, к которой стремится z, и бесконечность представляют собой две отдельные одноточечные орбиты — аттракторы для точек на комплексной плоскости. Граница между «зонами их притяжения» — это множество Жюлиа, невероятно тонкое и сморщенное. Оно тоже является орбитой, но не является аттрактором в строгом смысле слова. Точки, уже содержащиеся в этой пограничной области, совершают в ней беспорядочные скачки. Непосредственно вычислить множество Жюлиа непросто, потому что точность выполнения арифметических операций компьютером может не позволить точно указать точки, которые с самого начала должны находиться на границе, а когда начинается итерационный процесс, точность еще более снижается и итерируемые переменные уходят из поля зрения.

Как уже говорилось, каждому возможному значению параметра c соответствует свое, отличное от других множество Жюлиа. Множество Мандельброта в некотором смысле охватывает все возможные множества Жюлиа. Оно описывает судьбу переменной, выходящей из начала координат комплексной плоскости, для всех возможных значений параметра c. Для некоторых множеств Жюлиа область хаоса представляет собой лишь тонкую древовидную структуру или даже некую симметричную россыпь точек, область, как бы посыпанную перцем. Читатели, наверное, помнят, что такие множества Жюлиа соответствуют значениям c, находящимся на границе множества Мандельброта или за его пределами.

Недавно мы встретились с Пайтгеном на конференции в Асиломаре (шт. Калифорния). Когда мы прогуливались вдоль пляжа, разговаривая о том, о сем, он охарактеризовал множество Мандельброта как своего рода огромную книгу, в которой каждое множество Жюлиа — не более чем страничка. По тому положению, которое занимает точка c в множестве Мандельброта, можно предсказать поведение итерируемой переменной, характеризуя общую форму и размер сопутствующего множества Жюлиа. Здесь дело не только в том, будет ли оно связным или несвязным. Если, к примеру, выбрать c из перешейка между основной частью множества Мандельброта и одной из его «точек», то соответствующее множество Жюлиа окажется сжатым в структуру, также состоящую из перешейков и точек. Аналогия, согласно которой множество Мандельброта является как бы словарем для множеств Жюлиа, предполагает фундаментальное различие между этими множествами. Структура множества Жюлиа подобна самой себе (в различном масштабе), в то время как для множества Мандельброта (даже для его границы) это не так. В противном случае, как говорит Пайтген, множество Мандельброта не обладало бы способностью кодировать в себе несчетные количества родственных множеств Жюлиа.

Трехмерное поперечное сечение четырехмерного множества Мандельброта

В упомянутой книге «Красоты фракталов» есть еще много интересного. Я весьма признателен Пайтгену за те изображения, которые он предоставил для иллюстрации в настоящей статье. Не все они получены описанными здесь способами, однако рецепты изготовления всех изображений даны в книге.

В качестве заключительного замечания об исследованиях, ведущихся в настоящее время специалистами в области динамических систем, я упомяну еще об одном объекте, таящемся в пространстве высшей размерности, чудовище, порождаемом кубическими итерациями, формула которых, в отличие от формулы Мандельброта, содержит z^3 вместо z^2. Этот объект четырехмерный, его завивающиеся щупальца тянутся от главного «тела» и образуют причудливую картину, которую мы не способны увидеть. Однако ее трехмерные поперечные сечения могут быть вычислены и представлены подобно тому, как это сделано на рисунке на с. 92.

Поток писем со всех концов света по поводу статьи в августовском номере журнала, посвященной алгоритмическим головоломкам, не иссякает. Мы продолжаем разговор на эту тему, как и было обещано в предыдущей статье, в которой мы вернулись к алгоритмическим головоломкам. Прежде чем отправиться в пустыню и дать ответы к двум последним задачкам, сформулированным в первой статье об алгоритмических головоломках, я должен исправить ошибку, допущенную мною в рассуждениях о задачах с поездами. М. Блам, специалист по вычислительной математике из Калифорнийского университета в Беркли, заметил, что объем работы, требующейся, чтобы пропустить один поезд мимо другого по приведенному в статье алгоритму, на самом деле пропорционален n^3, а не n^2, как утверждалось в статье (напомним, что n — число вагонов поезда). Грубо говоря, каждый из n вагонов поезда перемещается на n единиц длины n раз. Бламу удалось придумать такой алгоритм, который решает ту же задачу при затратах работы, пропорциональных $n^2 \times \log n$. К сожалению, недостаток места не позволяет мне привести здесь алгоритм Блама.

По условиям первой задачи о «разведчике пустыни» автомашина могла везти в своем кузове цистерну бензина емкостью 50 галлонов и имела бензобак, вмещавший 10 галлонов. В прошлой статье было показано, что, имея на базе две цистерны горючего, автомашина может удалиться от базы на максимальное расстояние, равное 733,3 мили, прежде чем запас горючего будет полностью исчерпан.

Алгоритмы, представленные Ч. Новогорски из Нэплза (шт. Флорида) и Н. Рокком из Уинтерсвиля (шт. Огайо), свидетельствуют о том, что, имея три цистерны горючего, автомашина может удалиться от базы на расстояние 860 миль. Результаты большинства других читателей, попробовавших свои силы в этой задаче, были меньше этой величины. На самом деле даже общие формулы, представленные большинством читателей, дали более низкий результат, когда вместо n, количества цистерн в общем случае, было подставлено число 3. Я не могу поэтому поручиться за правильность формул, подобных той, которую прислал Л. Лейнвебер из Кливленда (шт. Огайо). Однако она является типичным представителем тех формул из числа представленных читателями, которые дают наилучшие результаты:

$$5\sum_{i=1}^{n}\frac{100}{2i-1}-\frac{100}{2n-1}+100$$

Здесь греческая буква сигма (Σ) означает суммирование последовательности величин $100/(2i-1)$, получающихся при различных i, принимающего значения от 1 до n. Формула означает, таким образом, что нужно сложить все n величин и полученную сумму помножить на 5.

Во второй задаче о разведчике пустыни автомашину можно было заправлять на n складах с горючим, расположенных в произвольных пунктах кругового маршрута, объезжаемого автомашиной. Количество горючего на каждом складе также произвольно, но горючего на всех складах должно быть ровно столько, сколько необходимо, чтобы машина могла полностью объехать свой маршрут (конечно, при условии, что она не начнет поездку от неподходящего склада). Ее собственный бензобак может вместить любое количество горючего, а в исходном состоянии он пуст. Нужно определить, откуда машина должна начать поездку.

Ряд читателей, включая А. Лавриджа из Лонг-Бича (шт. Калифорния), придумали очень интересный прием, при помощи которого можно решить эту задачу. Представим себе, что автомашина совершает «пробную» поездку, стартуя у произвольно выбранного склада и двигаясь в произвольном направлении. Попутно будем рисовать график количества бензина в баке, как функцию пройденного расстояния, и не будем останавливаться, даже если горючее полностью исчерпалось. В этом случае график просто перейдет в отрицательную область. После каждой заправки на складе горючего кривая резко поднимается вверх и затем начинает постепенно снижаться. В конце концов машина вернется к исходной точке. Теперь водитель должен проанализировать график и выбрать тот склад, при приближении к которому машина имела минимальное количество бензина (перед заправкой). От этого склада и следует начинать поездку.

Свойства хаоса были исследованы многочисленными читателями, пожелавшими и сумевшими запрограммировать итерационную формулу, приведенную в статье нашей рубрики в октябрьском номере журнала. На самом деле несколько энтузиастов, среди них Х. Марк из Сафферна (шт. Нью-Йорк), последовали моему предложению и проследили за самим итерационным процессом, наблюдая за паутиной, сплетающейся вокруг параболы. Сходимость процесса можно наблюдать по мере того, как эта паутина принимает стабильную форму. В отличие от этого в хаотическом режиме она заполняет целую область экрана запутанной структурой белых квадратов, не устанавливающихся ни в какую стабильную картину. Ч. Плантц из Уэст-Браунсвиля (шт. Пенсильвания) воспользовался своим персональным компьютером как микроскопом, нацелив его на дугу бифуркационной диаграммы, имеющую форму стремени, как раз над одной из хаотических областей. Но он обнаружил там не «соль и перец», как я, возможно, предсказал бы ему, а скорее слои и складчатые структуры, вторгающиеся в зоны хаоса.