ней дугами, покрашенными в цвет этой вершины, мы получим непересекающиеся подмножества S. Они не будут образовывать разбиения, так как не все покрашенные дуги в них входят. Но у нас в семействе есть одноэлементные множества, соответствующие всем покрашенным дугам, так что мы можем нужные из них добавить и получить разбиение. Таким образом, из существования раскраски вытекает существование разбиения. Наше утверждение доказано.

В этом месте читатель может спросить нас: ну хорошо, вы продемонстрировали, как одна конкретная задача сводится к другой. Но каким образом можно доказать, что всякая переборная задача сводится к данной, ведь их же бесконечно много? К сожалению, ответ на этот вопрос предполагает знакомство с математической логикой и теорией алгоритмов и выходит за рамки нашей книжки. Его можно найти в упоминавшейся книге Ахо, Хопкрофта, Ульмана. Заметим, что после того, как полнота какой-то одной переборной задачи доказана, можно доказать полноту другой задачи, сведя к ней первую. Например, известно, что задача о раскраске является полной, т. е. любая переборная задача к ней сводится. Поэтому приведенное выше рассуждение показывает, что и задача о разбиении является полной.

Однако, к счастью, не все переборные задачи являются полными. Для некоторых переборных задач удается придумать алгоритм, позволяющий избавиться от перебора, хотя часто это требует большой изобретательности. Мы приведем одну такую задачу вместе с ее решением.

З а д а ч а о р а с к р о е м н о г о у г о л ь н и к а. Дан выпуклый n-угольник. Его раскроем назовем разбиение его на треугольники с помощью $n-3$ непересекающихся диагоналей. Стоимостью раскроя называется суммарная длина проведенных диагоналей. Требуется найти наименьшую возможную стоимость раскроя.

Казалось бы, для ее решения нужно перебрать все способы раскроя и сравнить суммарную длину разрезов (диагоналей) в каждом из них. Для четырехугольника есть 2 раскроя (рис. 26), для пятиугольника — 5 раскроев (рис. 27), для шестиугольника — 14 раскроев (рис. 28) и т. д. Можно убедиться, что с ростом n число раскроев растет очень быстро и их перебор

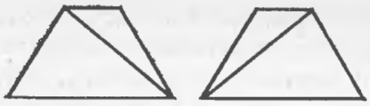

Рис. 26

Рис. 27

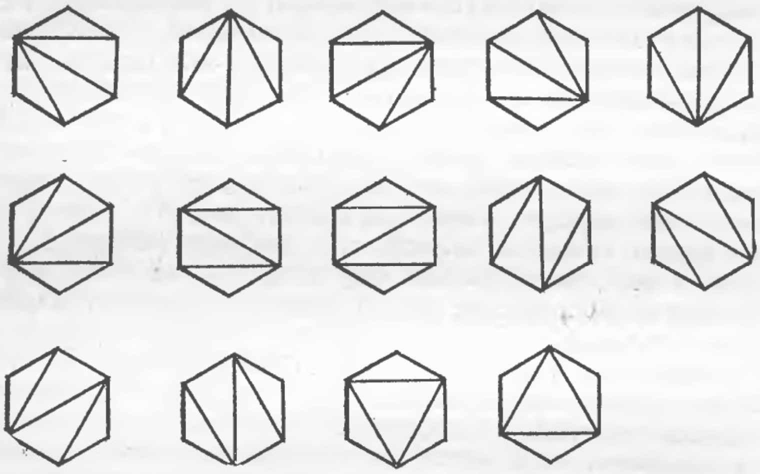

Рис. 28

скоро становится практически невозможным. Тем не менее задача может быть решена сравнительно просто.

Общая идея, которая оказывается здесь применимой,— «разделяй и властвуй» — сведение задачи к нескольким подзадачам того же типа, но меньшего размера. Проиллюстрируем экономию, достигаемую таким способом, на модельном примере. Зафиксируем произвольную диагональ многоугольника и будем временно рассматривать только раскрои, включающие данную диагональ. Чтобы найти среди них раскрой наименьшей стоимости, нужно соединить наилучшие раскрои отдельно для левой и правой частей многоугольника, на которые он разбивается этой диагональю. Выгода, доставляемая этим простым приемом, видна из такой простой оценки: если одна часть многоугольника допускает k раскроев, а другая — l, то всего раскроев (включающих эту диагональ) kl. Используя сказанное, мы ограничиваемся рассмотрением отдельно k и l раскроев, всего $k + l$, что может быть существенно меньше kl.

Таким способом можно найти наилучший раскрой, включающий данную диагональ. Сделав это для каждой из диагоналей многоугольника, можно найти наилучший его раскрой. Это соображение уже существенно помогает, если в нашем распоряжении есть много одинаковых исполнителей. Возьмем столько пар исполнителей, сколько диагоналей в нашем многоугольнике. Каждой паре дадим части многоугольника, на которые он разбивается одной из диагоналей. Когда они представят нам наилучшие раскрои частей, соединим их в раскрои всего многоугольника и найдем среди полученных раскроев наилучший. При этом каждый из исполнителей может решать поставленную перед ним задачу тем же способом, разбив ее на меньшие части и поручив их решение своим подчиненным и т. д. Такая организация работ позволит получить ответ довольно быстро, особенно если нам удастся добиться, чтобы число вершин многоугольников при каждом разбиении существенно уменьшалось (например, на одну треть). Попытайтесь понять, можно ли этого достичь.

Однако если исполнитель у нас один, указанная тактика не дает хорошего результата. Нам придется существенно изменить ее, рассматривая треугольники вместо диагоналей. Начнем с нескольких очевидных замечаний.

Всякая сторона многоугольника, разрезанного на треугольники непересекающимися диагоналями, является стороной некоторого треугольника раскроя (рис. 29). Стоимость раскроя, содержащего указанный треугольник, есть сумма стоимостей раскроев двух полученных многоугольников плюс длина двух проведенных диагоналей. Таким образом, чтобы найти стоимость самого дешевого раскроя многоугольника среди всех раскроев, содержащих этот

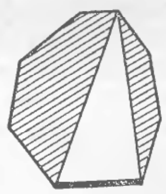

Рис. 29

треугольник, надо сложить стоимости наилучших раскроев полученных многоугольников и прибавить к ним длины диагоналей. А стоимость раскроя, являющегося наилучшим среди всех, получится, если взять минимум по всем треугольникам с данной стороной. (Возможен и случай, когда один из двух получающихся многоугольников вырождается в отрезок.)

Заметим, что при таком сведении раскроя многоугольника к раскрою двух других в этих двух многоугольниках одна сторона является диагональю исходного, а остальные — его сторонами, выделенная же первоначально сторона вовсе не входит в новые многоугольники. Это замечание, на первый взгляд выглядящее как не очень относящееся к делу, в действительности является решающим для построения эффективного алгоритма.

Объясняется это следующим. Мы указали, как найти стоимость наилучшего раскроя многоугольника, если известны стоимости наилучшего раскроя других, меньших. Чтобы извлечь максимальную выгоду из этого способа, нам понадобится таблица, содержащая сведения о стоимостях раскроя. Такая таблица будет приемлемой, лишь если ее объем (количество многоугольников, для которых мы находим и запоминаем стоимость их наилучшего раскроя) не слишком велик. На первый взгляд кажется, что необходимо иметь в таблице все многоугольники, вершинами которых являются некоторые из вершин исходного. А их много (порядка 2^n). Оказывается, однако, что можно ограничиться хранением сведений лишь о некоторых многоугольниках, которые мы будем называть специальными.

Многоугольник будем называть специальным, если его вершины являются вершинами исходного многоугольника, а все стороны, кроме, быть может, одной, — сторонами исходного. Будем решать задачу о раскрое специального многоугольника указанным способом, выбрав в качестве выделенной стороны диагональ исходного многоугольника. В этом случае получающиеся два многоугольника снова будут специальными и будут иметь меньшее число сторон. Таким образом, нас будут интересовать лишь стоимости наилучшего раскроя специальных многоугольников, а их не так много.

Сколько? Подсчитаем: каждая диагональ исходного многоугольника делит его на два специальных многоугольника. Так может быть получено $n(n-3)$ специальных многоугольников (число диагоналей равно $n(n-3)/2$), и остается учесть исходный многоугольник. Всего получается, следовательно, не более n^2 специальных многоугольников.

Построив таблицу, содержащую графу для стоимости наилучшего раскроя каждого специального многоугольника, начнем ее постепенно заполнять. Заполнение будем производить в порядке возрастания числа сторон многоугольников. Для треугольников стоимость раскроя равна 0 (резать нечего).

Допустим, мы уже заполнили таблицу для всех специальных многоугольников с не более чем k вершинами. Теперь наша задача состоит в том, чтобы найти наименьшую стоимость раскроя для специальных $(k+1)$-угольников. Нарисуем специальный $(k+1)$-угольник; выделим (жирной линией) сторону, не являющуюся стороной исходного многоугольника. Нам надо найти стоимость самого дешевого его раскроя. Все раскрои делятся на классы в зависимости от того, в какой треугольник раскроя входит выделенная сторона (рис. 30). Чтобы найти наименьшую стоимость

Рис. 30

раскроя, достаточно найти наименьшую стоимость раскроя в каждом из классов и затем выбрать минимальное среди этих чисел. Чтобы найти наименьшую стоимость раскроя в каждом классе, нужно сложить длины двух изображенных диагоналей со стоимостью раскроя двух возникающих многоугольников с меньшим числом сторон (рис. 31). Замечательным образом оказывается, что эти многоугольники также являются специальными, и стоимость их раскроя можно найти в таблице, так как число вершин в них меньше. Случай, когда один из них вырождается в отрезок, рассматривается аналогично.

Посмотрим, сколько действий нужно сделать для заполнения одной клетки таблицы. Нахождение мини-

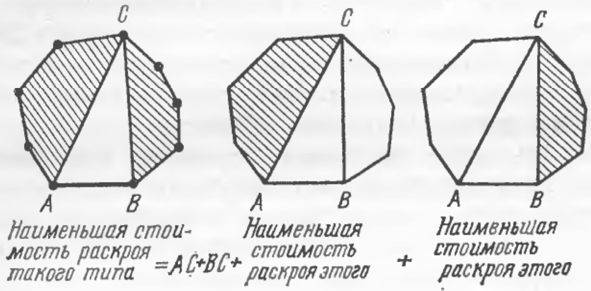

Рис. 31

мальной стоимости раскроя каждого типа (для данного треугольника), как мы видели, требует ограниченного числа действий (надо сложить два числа из таблицы и длины двух диагоналей). Типов раскроев не больше чем n (число вершин исходного многоугольника), поэтому общее число действий, необходимых для заполнения одной клетки таблицы, не превосходит $\mathrm{const}\cdot n$. А для заполнения всей таблицы необходимо не более $\mathrm{const}\cdot n^3$ действий. Это не так много, и задача остается практически разрешимой и для n порядка нескольких сотен.

ГЛАВА 15

ИГРЫ, ИГРЫ, ИГРЫ...

Эффектной демонстрацией «интеллектуальных» возможностей компьютеров являются программы, играющие в шахматы, шашки, кала́х и т. д. Поэтому особенно интересно понять, как такие программы устроены «внутри». К сожалению, объяснение этого выходит за рамки нашей книжки, и мы ограничимся изложением свойств игр, лежащих в основе таких программ.

Прежде всего скажем о том, какие игры мы будем рассматривать. Мы предполагаем, что в игре участвуют двое, что они ходят по очереди и что каждый игрок рас-