Алгоритмические головоломки

А. Левитин
М. Левитина

Лаборатория знаний

АЛГОРИТМИЧЕСКИЕ ГОЛОВОЛОМКИ

ANANY LEVITIN
MARIA LEVITIN

Algorithmic PUZZLES

OXFORD
UNIVERSITY PRESS

АНАНИЙ ЛЕВИТИН
МАРИЯ ЛЕВИТИНА

Алгоритмические ГОЛОВОЛОМКИ

2-е издание (электронное)

Перевод с английского
Ж. А. Меркуловой,
Н. А. Меркулова

Москва
Лаборатория знаний
2019

УДК 51-028.41+794
ББК 22.1я92
Л36

Левитин А.

Л36 Алгоритмические головоломки [Электронный ресурс] / А. Левитин, М. Левитина ; пер. с англ. Ж. А. Меркуловой, Н. А. Меркулова. — 2-е изд. (эл.). — Электрон. текстовые дан. (1 файл pdf : 328 с.). — М. : Лаборатория знаний, 2019. — Систем. требования: Adobe Reader XI ; экран 10".

ISBN 978-5-00101-643-4

Книга является уникальной коллекцией 150 головоломок, каждая из которых снабжена указанием и решением. Задачи сгруппированы в зависимости от уровня сложности. Издание дополнено двумя обучающими разделами по стратегиям разработки и анализа алгоритмов.

В настоящее время алгоритмические головоломки часто используются на собеседованиях при приеме на работу. Они призваны развить аналитическое мышление и просто разнообразить досуг.

Для всех любителей математики.

УДК 51-028.41+794
ББК 22.1я92

Деривативное электронное издание на основе печатного аналога: Алгоритмические головоломки / А. Левитин, М. Левитина ; пер. с англ. Ж. А. Меркуловой, Н. А. Меркулова. — 2-е изд. — М. : Лаборатория знаний, 2019. — 325 с. : ил. — ISBN 978-5-00101-188-0.

12+

ISBN 978-5-00101-643-4

Посвящается Максу с любовью

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ В ВОПРОСАХ И ОТВЕТАХ

О ЧЕМ ЭТА КНИГА?

Эта книга представляет собой сборник алгоритмических головоломок — головоломок, для решения которых требуются, в явном или неявном виде, чётко определённые процедуры решения задач. Книга представляет собой уникальный сборник таких головоломок. В неё включены как старые классические задачи, вошедшие в фольклор математиков и специалистов по информатике, так и новые головоломки, которые предлагают решить во время собеседования в крупных компаниях при приёме на работу.

У этой книги есть две цели:

- развлечь широкий круг читателей, интересующихся головоломками;
- способствовать развитию алгоритмического мышления на высоком уровне (без компьютерного программирования) с помощью ознакомления с основными стратегиями разработки алгоритмов и методами анализа, которые были тщательно отобраны авторами.

Алгоритмы — это краеугольный камень информатики, и программирование без них невозможно. Однако было бы неправильно, как это делают многие, ставить знак равенства между алгоритмом и компьютерной программой. Некоторые алгоритмические головоломки старше компьютеров более чем на тысячу лет. Однако следует признать, что распространение компьютеров сделало доступным решение алгоритмических задач во многих областях современной жизни, от естественных и гуманитарных наук до искусства и индустрии развлечений. Решение алгоритмических головоломок — это наиболее продуктивный и уж точно самый приятный способ развить и улучшить навыки алгоритмического мышления.

ДЛЯ КОГО ЭТА КНИГА?

Эта книга может быть интересна для трёх больших категорий читателей:

- любителей головоломок;
- тех, кто хочет развить алгоритмическое мышление, в том числе учителей и учащихся;

- тех, кто готовится к собеседованию при приёме на работу, во время которого предлагают головоломки, а также тех, кто проводит эти собеседования.

Любителям головоломок понравится этот сборник, содержащий головоломки разных типов и тем. Они встретят как головоломки, любимые во все времена, так и малоизвестные «жемчужины». Никаких знаний по информатике или даже интереса к ней не требуется. Читатель, не обладающий такими знаниями, может просто пропустить описание стратегий разработки и анализа алгоритмов в разделе «Решения».

В последнее время термин «алгоритмическое мышление» очень часто употребляется преподавателями по информатике, и не без оснований: повсеместное распространение компьютеров в современном мире действительно делает необходимым овладение навыками алгоритмического мышления практически для каждого студента. Головоломки — идеальное средство для приобретения этих важных навыков в силу двух причин. Во-первых, головоломки — это занятно и человек обычно готов приложить больше усилий для решения головоломок, чем для решения обычных скучноватых упражнений. Во-вторых, алгоритмические головоломки заставляют мыслить на более абстрактном уровне. Даже студенты, обучающиеся информатике, чаще думают о решении алгоритмических задач в терминах компьютерного языка, который они знают, и не пытаются применить общие стратегии разработки и анализа алгоритмов. Головоломки помогают восполнить этот существенный недостаток.

Головоломки в этой книге, безусловно, можно использовать для самостоятельного обучения. Вместе с обучающим материалом они, на наш взгляд, позволяют ознакомиться с основными алгоритмическими понятиями и методами. Головоломки могут использовать и преподаватели компьютерных курсов как в университетах, так и в средней школе, в качестве вспомогательных упражнений и материала для проектов. Эта книга также может быть интересна для обучения на курсах по принятию решений, особенно на тех курсах, которые основаны на решении головоломок.

Для людей, готовящихся к собеседованию, эта книга будет полезна по двум причинам. Во-первых, в ней есть много примеров головоломок, с которыми они могут встретиться, с полным решением и комментариями. Во-вторых, в книге содержится краткий учебный материал по стратегиям разработки и методам анализа алгоритмов. По словам менеджеров, предлагающих на собеседовании решить головоломки, для них в большей степени важно увидеть правильный подход к решению головоломки, а не само решение. На потенциального работодателя произведёт очень хорошее впечатление, если вы продемонстрируете ваш опыт в применении общих стратегий разработки алгоритмов.

КАКИЕ ГОЛОВОЛОМКИ ВКЛЮЧЕНЫ В КНИГУ?

Алгоритмические головоломки составляют малую часть среди тысяч математических головоломок, придуманных на протяжении многих лет. Подбирая головоломки для этой книги, мы руководствовались следующими критериями.

Во-первых, мы хотели, чтобы головоломки иллюстрировали некоторые общие принципы разработки и анализа алгоритмов.

Во-вторых, мы хотели, чтобы они были красивыми и элегантными, хотя эти качества, конечно, субъективны.

В-третьих, мы хотели, чтобы головоломки были различного уровня сложности. Сложность головоломки трудно точно определить; иногда головоломки, которые легко решают ученики средней школы, ставят в тупик профессоров математики. Тем не менее, мы поделили нашу книгу на три раздела — головоломки лёгкие, средней сложности и сложные — чтобы помочь читателю оценить уровень головоломки. В пределах каждого раздела мы также отсортировали головоломки по уровню сложности. Для решения головоломок из раздела «Лёгкие головоломки» нужна только математика средней школы. Для решения некоторых задач из следующих двух разделов применяется метод индукции, но в общем, чтобы решить все головоломки в книге, будет достаточно школьной математики. Кроме того, во второй части учебного раздела вкратце освещаются такие темы, как бинарные числа и простые рекуррентные соотношения. Конечно, это не означает, что все головоломки в книге — лёгкие. Некоторые из них — особенно в конце последнего раздела — являются по-настоящему сложными. Но их трудность вовсе не в том, что они требуют сложной математики, поэтому они не должны отпугивать читателя.

В-четвёртых, мы чувствовали себя обязанными включить несколько головоломок, имеющих историческое значение.

В заключение отметим, что мы включили в книгу головоломки только с ясной постановкой задачи и решением, избегая таких трюков, как преднамеренная двусмысленность, игра слов и т. д.

Нужно сделать ещё одно замечание. Многие головоломки из этой книги можно решить с помощью полного перебора или поиска с возвратом (эти стратегии описаны в первой части учебного раздела). Однако, это *не тот* подход, которым нужно наслаждаться при решении этих головоломок. Он требуется только в случаях, когда это специально указано. Поэтому мы исключили такие категории головоломок, как судоку и криптарифмы, которые можно решить или полным перебором (перебором с возвратом), или же благодаря гениальной проницательности и озарению. Мы также решили не включать головоломки, относящиеся к физическим объектам, которые не очень легко описать, такие как Китайские кольца и кубик Рубика.

ПОДСКАЗКИ, РЕШЕНИЯ И КОММЕНТАРИИ

В книге для каждой головоломки приведены подсказка, решение и комментарии. Книги по головоломкам редко включают подсказки, но мы считаем их ценным дополнением. Они могут слегка подтолкнуть в правильном направлении, оставляя, тем не менее, шанс читателю самому решить головоломку. Все подсказки собраны в отдельном разделе.

К каждой головоломке приведено решение. Как правило, оно начинается с короткого ответа. Это сделано для того, чтобы дать читателю последнюю возможность самому решить головоломку: если ответ читателя не тот, что в решении, он может прекратить чтение решения и попытаться снова решить головоломку.

Алгоритмы описаны без использования специального формата или псевдокода. Акцент делается на идеях, а не на незначительных деталях. Переписать решения в более формальном виде — это уже само по себе полезное упражнение.

Большинство комментариев обращают внимание на общую идею алгоритма, которую демонстрируют головоломка и её решение. Иногда в комментариях также есть ссылки на похожие головоломки из этой книги или из других источников.

В большинстве книг, посвящённых головоломкам, не указывается происхождение головоломки. Обычно это объясняется тем, что пытаться найти автора головоломки — всё равно, что пытаться найти автора шутки. Хотя в этом и есть значительная доля правды, мы решили упомянуть о самых ранних источниках головоломок, которые нам известны. Однако читатель должен иметь в виду, что мы не производили хоть сколько-нибудь серьёзного поиска источников головоломок. Если бы мы это сделали, получилась бы совсем другая книга.

ЧТО ПРЕДСТАВЛЯЕТ СОБОЙ УЧЕБНЫЙ РАЗДЕЛ?

В книгу включён учебный раздел, где на примерах головоломок описаны общие стратегии разработки и методы анализа алгоритмов. Хотя почти все головоломки в книге можно решить без малейшего знания тем, освещённых в учебном разделе, вне сомнения он поможет решить головоломки гораздо легче и, что важно, с большей пользой. Кроме того, в решениях, комментариях и некоторых подсказках используется та же терминология, объяснение которой даётся в учебном разделе.

Учебный раздел написан на максимально доступном уровне, чтобы быть понятным широкому кругу читателей. Если читатель — специалист по информатике, то он вряд ли найдёт там для себя что-то новое, разве что примеры головоломок. В то же время, такой читатель может быстро освежить свои знания фундаментальных идей разработки и анализа алгоритмов.

ПОЧЕМУ В КНИГЕ ДВА УКАЗАТЕЛЯ?

В дополнение к стандартному указателю в книге есть ещё указатель, где головоломки сгруппированы по стратегии разработки или типу анализа соответствующего алгоритма. Этот указатель призван помочь читателю соотнести задачу с конкретной стратегией или методом решения, тем самым являясь дополнительной подсказкой.

Мы надеемся, что читатели найдут книгу как занимательной, так и полезной. Мы также надеемся, что они разделят наше восхищение красотой и удивительной человеческой изобретательностью, которые видны во многих головоломках этой книги.

Ананий Левитин, Мария Левитина
Май 2011 г.
algorithmicpuzzles.book@gmail.com

БЛАГОДАРНОСТИ

Мы хотели бы выразить нашу глубокую благодарность рецензентам книги: Тиму Шартье (Колледж Дэвидсона), Стивену Лукасу (Университет Джеймса Мэдисона) и Лоре Таалман (Университет Джеймса Мэдисона). Их горячая поддержка идеи книги и конкретные предложения по её содержанию были, безусловно, нам очень полезны.

Мы также благодарны Симону Берковичу из Университета Джорджа Вашингтона за обсуждение тем головоломок и за чтение части рукописи книги.

Мы благодарны всем сотрудникам издательства *Oxford University Press* и их коллегам, которые работали над книгой. Мы особенно благодарны нашему редактору Филлис Коэн за её неустанные усилия сделать книгу лучше. Мы также благодарим помощника редактора Халли Стеббинс, дизайнера обложки Наталью Балнову и менеджера по продажам Мишель Келли. Мы ценим работу Ричарда Кампа, редактора, подготовившего рукопись к печати, а также усилия Дженифер Коуинг и Кирана Кумара, которые отвечали за выпуск книги.

СПИСОК ГОЛОВОЛОМОК

ГОЛОВОЛОМКИ УЧЕБНОГО РАЗДЕЛА

Данный список содержит все головоломки, включённые в учебный раздел. Головоломки перечислены в том порядке, в котором они появляются в книге. Номер страницы указывает на ту страницу, где описана сама головоломка, а их решения даны непосредственно в учебном разделе, следом за формулировкой.

ГОЛОВОЛОМКИ ОСНОВНОГО РАЗДЕЛА

В этом списке указаны все 150 головоломок, включённых в основной раздел книги. Три номера после головоломки указывают на страницы, где приведены условие, подсказка и решение соответственно.

ГОЛОВОЛОМКА В КАЧЕСТВЕ ЭПИГРАФА: КТО ЭТО СКАЗАЛ?

Определите, какому автору из перечисленных ниже принадлежат эти цитаты.

Человек с молотком подходит к каждой проблеме, как к забиванию гвоздей. Большой молоток нашего века — это алгоритм.

Решение задач — это практический навык, как, скажем, плавание. Все практические навыки мы приобретаем с помощью подражания и практики.

Нет лучшего способа развеять скуку, как ввести в курс занимательные темы и элементы развлечения — игру, юмор, красоту и сюрприз.

Не само знание, но процесс обучения, не обладание, но движение к цели, — вот что гарантирует настоящее наслаждение.

Если я случайно упустил что-нибудь более или менее уместное или необходимое, я прошу снисходительности, поскольку нет ни одного человека, который совершенно безупречен и предусмотрителен по отношению ко всем вещам.

Уильям Паундстоун,
автор книги *«Как сдвинуть гору Фудзи? Подходы ведущих мировых компаний к поиску талантов»*.

Дьёрдь Пойа (1887–1985),
выдающийся венгерский математик, автор книги *«Как решать задачу»*, классической книги по решению задач.

Мартин Гарднер (1914–2010),
американский писатель, особенно хорошо известный по рубрике «Математические игры» в журнале *Scientific American* и книгам по занимательной математике.

Карл Фридрих Гаусс (1777–1855),
великий немецкий математик.

Леонардо Пизанский (Фибоначчи) (1170 – ок. 1250),
выдающийся итальянский математик, автор *Liber Abaci* (*«Книга абака»*), одной из важнейших книг по математике в истории.

Глава 1

УЧЕБНЫЙ РАЗДЕЛ

ОБЩИЕ СТРАТЕГИИ РАЗРАБОТКИ АЛГОРИТМОВ

Цель данного раздела — дать краткий обзор общих стратегий разработки алгоритмов. Хотя каждая из этих стратегий применима не ко всем головоломкам, все вместе они представляют собой мощный набор средств. Неудивительно, что эти стратегии используются также для решения многих задач в информатике. Таким образом, обучение тому, как применять эти стратегии к решению головоломок, может служить прекрасным введением в эту важную область.

Прежде чем мы приступим к рассмотрению основных стратегий разработки алгоритмов, нужно сделать важное замечание о двух типах алгоритмических головоломок. В каждой головоломке есть входные данные, по которым определяется, о каком *случае* задачи идёт речь. Случай может быть либо частным (например, найти фальшивую монету среди восьми монет с помощью взвешивания) или общим (например, найти фальшивую монету среди n монет путём взвешивания). Если читатель имеет дело с частным случаем головоломки, он не обязан решать её для общего случая. Иногда другие случаи имеют другое решение или даже вообще не имеют решения. Но с другой стороны, конкретное число в формулировке головоломки может вовсе не иметь никакого значения. Тогда решение головоломки для общего случая не только принесёт большее удовлетворение, но может оказаться проще. Но, независимо от того, задана ли головоломка для частного случая или в общей форме, всегда полезно решить её для нескольких простых случаев. Иногда это может навести читателя на неверный путь, но гораздо чаще будет способствовать более глубокому пониманию задачи.

Полный перебор

Теоретически многие головоломки можно решить с помощью полного перебора — стратегии решения задач, при которой в условие задачи просто подставляются все возможные варианты решения, пока решение не будет найдено. При применении полного перебора обычно не нужно много изобретательности. Поэтому человеку (в отличие от компьютера) редко предлагают решить головоломку таким способом. При применении полного перебора самым главным ограничением является его неэффективность: как правило, число возможных вариантов, которые надо перебрать, растёт по меньшей мере экспоненциально с увеличением размера задачи, тем самым делая такой подход нецелесообразным не только для человека, но и для компьютера. В качестве примера рассмотрим задачу построения *магического квадрата* третьего порядка.

?	?	?
?	?	?
?	?	?

Рис. 1.1. Таблица 3×3, которую нужно заполнить целыми числами от 1 до 9, чтобы получился магический квадрат

Магический квадрат. Заполните таблицу 3×3 девятью различными целыми числами от 1 до 9 так, чтобы сумма чисел в каждой строке, каждом столбце и двух главных диагоналях была одинакова (рис. 1.1).

Сколькими способами можно заполнить такую таблицу? Допустим, мы ставим за один раз одно число, сначала поставим где-нибудь 1, а в самом конце — 9. Есть девять способов поставить 1, восемь способов поставить 2 и т. д.; последнюю цифру 9 можно поставить в единственную оставшуюся свободной ячейку в таблице. Таким образом, существует $9! = 9 \cdot 8 \cdot \ldots \cdot 1 = 362\,880$ способов разместить девять чисел в таблице 3×3. (Мы только что использовали стандартное обозначение $n!$, называемое *n факториал*, для произведения последовательных целых чисел от 1 до n.) Таким образом, если решать эту задачу путём полного перебора, пришлось бы расставить числа от 1 до 9 в таблице 362 880 возможными способами и для каждого случая проверить, не является ли сумма в каждой строке, столбце и главной диагонали одинаковой. Такой объём работы вне сомнения невозможно сделать вручную.

На самом деле, нетрудно решить эту головоломку, доказав сначала, что значение суммы равняется 15 и что нужно поместить 5 в центральную ячейку (см. головоломку «*Ещё раз о магическом квадрате*», № 29 в основном разделе книги). Кроме того, можно воспользоваться несколькими известными алгоритмами построения магических квадратов произвольного порядка $n \geqslant 3$, что особенно эффективно для нечётных n (см., например, [Pic02]). Конечно же, эти алгоритмы не основаны на полном переборе: количество вариантов решения, которое нужно проверить перебором, становится непомерно большим уже для n, равного всего лишь 5. Действительно, $(5^2)! \cong 1,5 \cdot 10^{25}$ и, следовательно, компьютеру, который делает 10 триллионов операций в секунду, понадобится около 49 000 лет, чтобы закончить работу.

Поиск с возвратом

В методе полного перебора есть две основные проблемы. Первая заключается в механизме определения всех возможных вариантов решения. Для некоторых задач мы можем составить хорошо структурированный набор вариантов. Например, варианты расстановки первых девяти положительных целых чисел в ячейках таблицы 3×3 (см. выше пример «*Магического квадрата*») можно получить с помощью перестановок этих чисел; для поиска же всех перестановок известно несколько алгоритмов. Однако во многих задачах варианты решения не образуют такую регулярную структуру. Вторая, и более сложная, проблема заключается в количестве вариантов, которые нужно проверить. Как правило, это количество растёт по крайней мере экспоненциально с увеличением размера задачи. Поэтому полный перебор разумно применять только для задач маленького размера.

Поиск с возвратом гораздо лучше по сравнению с примитивным «лобовым» подходом полного перебора. Это удобный метод сгенерировать возможные варианты решения, при этом исключая ненужные варианты. Главная его идея состоит в том, чтобы построить варианты решения по принципу «один компонент за один раз» и оценить такие частично построенные варианты решения следующим образом: если такой частично построенный вариант не нарушает ограничений задачи, то далее подбирается первый из допустимых вариантов для следующего компонента решения. Если для следующего компонента нет допустимого варианта, то не нужно рассматривать варианты ни для какого из оставшихся компонентов. В таком случае алгоритм возвращается на предыдущий шаг и заменяет последний компонент частично построенного решения на следующий из возможных вариантов.

Поиск с возвратом, в принципе, предполагает отсечение определённого количества неверных вариантов — чем больше это количество, тем быстрее алгоритм находит решение. В худшем варианте поиск с возвратом может в итоге сгенерировать такое же количество вариантов решения, как и полный перебор, но это происходит редко.

Поиск с возвратом можно интерпретировать как процесс построения дерева, на котором видны принимаемые решения. В компьютерных науках термин «дерево» используется для описания иерархических структур (например, генеалогическое древо, организационные схемы). *Дерево* обычно изображают следующим образом: *корень* (единственная узловая точка без *родителей*) сверху, а его *листья* (узлы без *детей*) — внизу диаграммы. Это просто удобное общепринятое изображение. Для перебора с возвратом такое дерево называется *деревом пространства состояний*. Корень этого дерева соответствуют началу процесса построения решения; мы считаем, что корень — нулевой уровень дерева. Дети корня — на первом уровне дерева — соответствуют возможным вариантам выбора первого компонента решения (например, ячейка в магическом квадрате содержит 1). Их дети — узлы на втором уровне — соответствуют возможным вариантам следующего компонента решения, и т. д. Листья могут быть двух видов. Первый вид — *бесперспективные узлы*, или *тупики* — представляют частично построенный вариант, который не ведёт к окончательному решению. Установив, что данный узел бесперспективен, алгоритм перебора с возвратом закрывает его (часть дерева обрезается), отменяет решение о последнем компоненте, возвращается обратно к *родителю* бесперспективного узла и рассматривает другой вариант выбора для этого компонента. Второй вид листьев даёт решение задачи. Если достаточно одного решения, алгоритм останавливается. Если нужны ещё решения, алгоритм продолжает поиск, возвращаясь к родителю листа.

Следующий пример является классическим для иллюстрации применения поиска с возвратом для конкретной задачи.

Задача об *n* ферзях. Разместите n ферзей на шахматной доске $n \times n$ таким образом, чтобы ни один из них не находился под боем другого. (Один ферзь бьёт другого, если находится с ним на одной линии по вертикали, горизонтали или диагонали.)

При $n = 1$ задача имеет тривиальное решение, и легко увидеть, что при $n = 2$ и $n = 3$ решения нет. Поэтому рассмотрим задачу о 4 ферзях и решим её с помощью поиска с возвратом. Поскольку каждого из четырёх ферзей нужно поместить на свою вертикаль, всё, что нам необходимо сделать, — определить для каждого ферзя клетку по горизонтали шахматной доски, показанной на рис. 1.2.

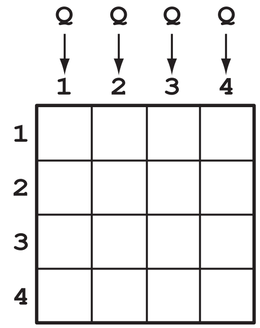


Рис. 1.2. Задача о 4 ферзях на шахматной доске

Начнём с пустой доски и поставим ферзя 1 в первую возможную позицию — позицию на горизонтали 1 и на вертикали 1. Затем ставим ферзя 2: после неуспешных попыток поставить его на горизонталях 1 и 2 и на второй вертикали помещаем его в первую возможную позицию — на клетку (3, 2), т. е. клетку на горизонтали 3 и на вертикали 2. Это оказывается тупиком, поскольку для ферзя 3 нет подходящей позиции на вертикали 3. Поэтому алгоритм возвращается назад и размещает ферзя 2 в следующую возможную позицию (4, 2). Затем ферзь 3 ставится на клетку (2, 3), что также оказывается тупиком. Алгоритм возвращается назад на все шаги к ферзю 1 и размещает его в позицию (2, 1). Ферзь 2 затем идёт на (4, 2), ферзь 3 на (1, 3) и ферзь 4 на (3, 4), что даёт решение задачи. Дерево пространства состояний данного поиска показано на рис. 1.3.

Если необходимо найти другие решения (а для задачи о 4 ферзях есть ещё одно решение), алгоритм просто возобновляет свои шаги с того листа дерева состояний, на котором он был остановлен. В качестве альтернативы для поиска второго решения в данном случае можно использовать симметричность доски.

Насколько быстрее можно найти решение с помощью поиска с возвратом по сравнению с полным перебором? Количество всех возможных позиций для четырёх ферзей на четырёх различных клетках доски 4×4 равно

$$\frac{16!}{4!(16-4)!} = \frac{16 \cdot 15 \cdot 14 \cdot 13}{4 \cdot 3 \cdot 2} = 1820.$$

Это общая формула, описывающая количество способов, которыми можно выбрать k различных элементов (неважно, в каком порядке) из множества n различных элементов. Это количество называется *сочетанием* из *n по k*, обозначается $\binom{n}{k}$ или $C(n,k)$ и равно $\frac{n!}{k!(n-k)!}$.

Если мы рассмотрим размещение ферзей только в различных колонках доски, общее количество возможных решений уменьшается до $4^4 = 256$. Если мы ещё добавим ограничение, что ферзи должны быть и в различных позициях по

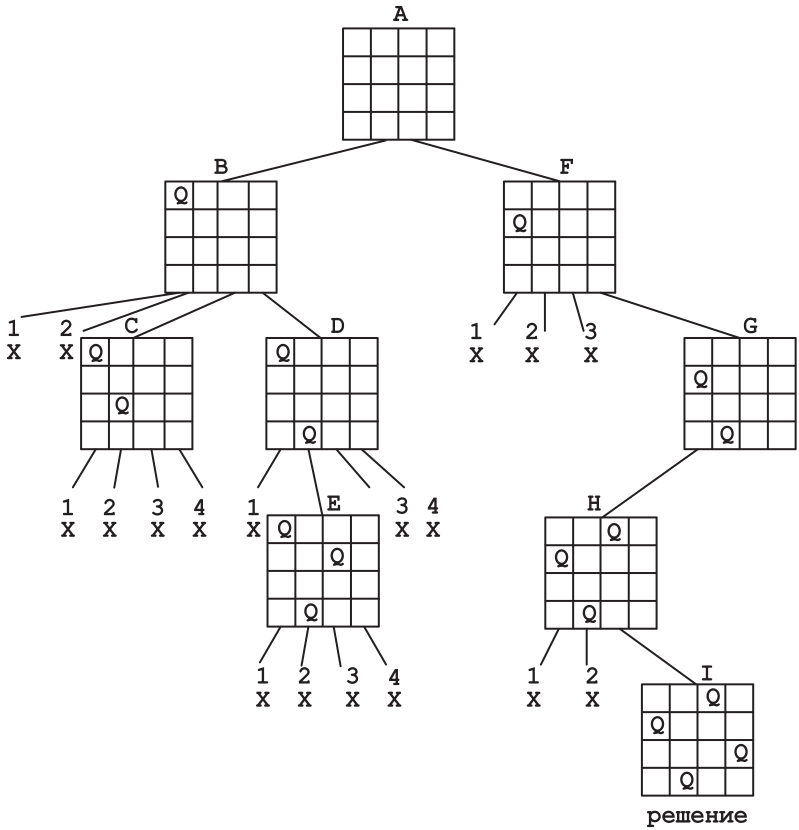


Рис. 1.3. Дерево пространства состояний, показывающее решение задачи о 4 ферзях (ферзи обозначены буквой Q) методом поиска с возвратом. Буква X обозначает неудачные попытки поместить ферзя на указанной вертикали. Буквы над узлами показывают, в какой последовательности генерируются узлы

горизонтали, количество вариантов уменьшается до $4! = 24$. Это количество вполне приемлемо, но не для более сложных частных случаев. Например, для обычной шахматной доски 8×8 количество возможных решений $8! = 40\,320$.

Читателю может быть интересно, что общее количество различных решений задачи о 8 ферзях равно 92, 12 из них качественно различны, а 80 других получаются из основных двенадцати с помощью поворотов и отражений. Что касается общей задачи об n ферзях, она имеет решение для любого $n \geqslant 4$; однако подходящей формулы, описывающей количество решений для любого n, не найдено. Известно, что количество решений очень быстро растёт с увеличением n. Например, количество решений при $n = 10$ равно 724, из них 92 качественно различаются, а при $n = 12$ эти количества равны 14 200 и 1787 соответственно.

С помощью поиска с возвратом можно решить многие головоломки из этой книги. Однако для каждой из них есть более эффективный алгоритм, который и должен найти читатель. В частности, для решения головоломки «*Снова задача об n ферзях*» (№ 140) из основного раздела книги нужно найти гораздо более быстрый алгоритм.

Уменьшай и властвуй

Стратегия «*уменьшай и властвуй*» *(метод уменьшения размера задачи — прим. ред.)* основана на нахождении соотношения между решением данной задачи и решением её более простого частного случая. Когда это соотношение найдено, оно естественным образом приводит к рекурсивному алгоритму, который последовательно сводит задачу ко всё более и более простым частным случаям до тех пор, пока частный случай не становится простым настолько, что его можно решить.[1] Приведём пример.

Задача о знаменитости. Знаменитость в группе из n людей — это человек, который не знает никого, но которого знают все. Задача — найти знаменитость, задавая людям один вопрос «Ты знаешь этого человека?»

Пусть, для простоты, нам известно, что знаменитость находится в группе из n людей. Задача может быть решена следующим образом с помощью алгоритма «уменьшить на единицу». Если $n = 1$, то этот единственный человек — знаменитость просто по определению. Если $n > 1$, выбираем двух человек из группы, например A и B, и спрашиваем A, знает ли он B. Если A знает B, удаляем A из группы людей, которые могут быть знаменитостью. Если A не знает B, удаляем B из группы. Затем решаем задачу рекурсивно (т. е. тем же методом) для оставшейся группы из $n - 1$ человек.

[1] Понятие *рекурсии* является одним из самых важных в информатике. Если читатель не знаком с ним, он может найти достаточно информации, например, в Википедии, по ссылкам в статье «Рекурсия (информатика)».

В качестве лёгкого упражнения читатель может решить головоломку «*Отряд солдат*» (№ 4) в основном разделе книги.

В принципе, более простые частные случаи в стратегии «*уменьшай и властвуй*» необязательно должны получаться при уменьшении n на 1. Хотя «уменьшить на единицу» это наиболее распространённый метод уменьшения, есть примеры уменьшения и на большее число. Можно получить гораздо более быстрый алгоритм, если нам удастся на каждом шаге уменьшать размер в постоянное число раз, например вдвое. Хорошо известный пример этого алгоритма применяется в следующей игре.

Угадай число (двадцать вопросов). Угадать число, находящееся в пределах от 1 до n, задавая вопросы, на которые можно ответить «да» и «нет».

Наиболее быстрый алгоритм для решения этой задачи — задавать вопрос, ответ на который уменьшает диапазон чисел, содержащих задуманное число, примерно в два раза на каждом шаге. Например, первый вопрос может быть, больше ли задуманное число, чем $\left\lceil \frac{n}{2} \right\rceil$ (это принятое обозначение для $\frac{n}{2}$, округлённого до ближайшего целого числа)[1]. Если ответ «нет», то задуманное число находится между целыми числами 1 и $\left\lceil \frac{n}{2} \right\rceil$. Если ответ «да», то задуманное число — между целыми числами $\left\lceil \frac{n}{2} \right\rceil + 1$ и n. В любом случае алгоритм уменьшает задачу размера n до частного случая этой же задачи размером примерно вдвое меньше изначальной. Повторяем этот шаг до тех пор, пока размер задачи уменьшится до 1, и задача решена.

Поскольку этот алгоритм уменьшает размер частного случая задачи (диапазон чисел, в которых содержится задуманное число) примерно в два раза на каждом шаге, он работает потрясающе быстро. Например, при $n = 1\,000\,000$ для алгоритма нужно не более 20 вопросов! Ещё быстрее был бы алгоритм, который уменьшает размер задачи в большее число раз, например в три раза.

[1] Число $\lceil x \rceil$ называется потолком действительного числа x и равняется наименьшему целому числу, которое больше или равно x. Например, $\lceil 2{,}3 \rceil = 3$; $\lceil 2 \rceil = 2$. Число $\lfloor x \rfloor$ называется полом x и равняется наибольшему целому числу, меньшему или равному x. Например, $\lfloor 2{,}3 \rfloor = 2$; $\lfloor 2 \rfloor = 2$.

Головоломка «*Фальшивая монета из восьми*» (№ 10) из основного раздела книги иллюстрирует стратегию «уменьшай в постоянное число раз», вариант стратегии «уменьшай и властвуй». Эта головоломка может служить хорошим упражнением.

Нужно отметить, что иногда проще установить соотношение между бо́льшим и меньшим случаями в обратном порядке. Это означает, что нужно сначала решить головоломку для самого маленького частного случая, потом следующего случая и т. д. Этот метод иногда называют «подходом по возрастающей». Примером является первое решение головоломки «*Разделение прямоугольника*» (№ 3) в основном разделе книги.

Разделяй и властвуй

Стратегия «*разделяй и властвуй*» *(метод декомпозиции — прим. ред.)* заключается в том, чтоб разделить задачу на несколько более лёгких подзадач (обычно такого же или похожего типа и желательно одного размера), решить каждую из них и, если это необходимо, скомбинировать их решения, чтобы получить решение исходной задачи. Эта стратегия лежит в основе многих эффективных алгоритмов, использующихся для решения важных задач в информатике. Удивительно, но с помощью алгоритмов «разделяй и властвуй» можно решить не так уж много головоломок. Приведём, однако, хорошо известный пример, который идеально демонстрирует эту стратегию.

Головоломка «Тримино». Замостить доску размером $2^n \times 2^n$, у которой нет одной клетки, угловыми тримино, т. е. фигурками L-образной формы, состоящими из трёх примыкающих друг к другу квадратов. Отсутствовать может любая из клеток доски. Тримино должны покрыть все клетки, кроме отсутствующей, без наложения друг на друга.

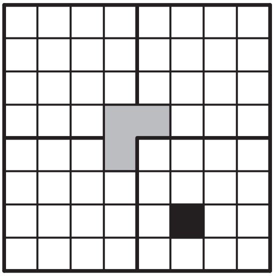


Рис. 1.4. Первый шаг алгоритма «разделяй и властвуй» в задаче покрытия фигурками тримино доски размером $2^n \times 2^n$, на которой отсутствует одна клетка

Задача может быть решена при помощи рекурсивного алгоритма «разделяй и властвуй». Поместим одно тримино в центр доски таким образом, что задача для случая n упрощается до четырёх случаев $n-1$ (рис. 1.4).

Алгоритм останавливается после того, как каждая часть доски размером 2×2 с одной отсутствующей клеткой покрыта одним тримино.

С помощью этого алгоритма читатель может в качестве быстрого, но полезного упражнения завершить покрытие доски размером 8×8, изображённой на рис. 1.4.

Большинство алгоритмов «разделяй и властвуй» решают меньшие подзадачи рекурсивно, поскольку, как в приведённом выше примере, они представляют собой меньшие частные случаи той же задачи. Однако так бывает не всегда. Для решения некоторых задач о досках, например, доску нужно поделить на части («поддоски»), которые необязательно меньшие версии изначальной. В качестве примеров решите головоломки «*Задача о 2n шашках*» (№ 37) и «*Замощение прямыми тримино*» (№ 78) из основного раздела книги.

О стратегии «разделяй и властвуй» необходимо сделать ещё одно замечание. Хотя некоторые считают стратегию «уменьшай и властвуй» (которая описана выше) особым случаем «разделяй и властвуй», лучше рассматривать её как отдельную стратегию. Принципиальная разница между ними заключается в количестве меньших подзадач, которые нужно решить на каждом шаге: несколько подзадач в алгоритмах типа «разделяй и властвуй» и всего одну в алгоритмах «уменьшай и властвуй».

Преобразуй и властвуй

«Преобразуй и властвуй» — это широко известный подход к решению задач, который основан на идее преобразования. Задача решается в два этапа. Сначала, на этапе преобразования, задача модифицируется или преобразуется в другую задачу, которую легче решить в силу тех или иных причин. На следующем этапе «властвования» она решается. В области решения задач с помощью алгоритмов можно выделить три варианта этой стратегии. Первый вариант называется *упрощение частного случая*; при этом задача решается путём преобразования исходного частного случая в другой частный случай той же задачи, который обладает некоей особенностью, позволяющей решить задачу легче. Второй вариант, называемый *изменение представления*, основан на преобразовании входа задачи в более подходящий для эффективного решения с помощью алгоритма. Третьим вариантом стратегии преобразования является *упрощение задачи*, при котором частный случай исходной задачи преобразуется в частный случай другой задачи.

В качестве первого примера приведём задачу-головоломку из книги Джона Бентли «Жемчужины программирования» [Ben00, p. 15–16].

Поиск анаграмм. Анаграммы — это слова, состоящие из одних и тех же букв. Например, слова «воз» и «зов», «атлас» и «салат». Нужно разработать алгоритм для поиска всех анаграмм в большом списке слов.

Эффективный алгоритм для решения этой задачи работает в две стадии. Сначала он присваивает каждому слову «ключ», состоящий из букв этого слова, расположенных в алфавитном порядке (изменение представления), а затем сортирует полученные «ключи» по алфавиту (сортировка данных относится к упрощению частного случая). Таким образом, анаграммы в полученном списке стоят рядом.

В качестве упражнения читателю предлагается решить головоломку «*Расстановка чисел*» (№ 43), основанную на этой же идее.

Другой пример изменения представления, который иногда бывает полезен, — это представить вход задачи в двоичной или троичной системе счисления. Если читатель не знаком с этой важной темой, дадим краткие пояснения. В десятичной позиционной системе, которую бо́льшая часть человечества использует на протяжении последних восьмисот лет, целое число представляется в виде комбинации степеней числа 10. Например, $1069 = 1 \cdot 10^3 + 0 \cdot 10^2 + 6 \cdot 10^1 + 9 \cdot 10^0$. В двоичной и троичной системах число представляется в виде комбинации степеней чисел 2 и 3 соответственно. Например, в двоичной системе $1069_{10} = 10000101101_2$, потому что $1069 = 1{\cdot}2^{10}+0{\cdot}2^9+0{\cdot}2^8+0{\cdot}2^7+0{\cdot}2^6+1{\cdot}2^5+0{\cdot}2^4+1{\cdot}2^3+1{\cdot}2^2+0{\cdot}2^1+1{\cdot}2^0$. А в троичной системе $1069_{10} = 1110121_3$, поскольку $1069 = 1 \cdot 3^6 + 1 \cdot 3^5 + 1 \cdot 3^4 + 0 \cdot 3^3 + 1 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0$. Десятичное число записывается десятью цифрами (от 0 до 9), двоичное число — двумя (0 и 1), троичное число — тремя (0, 1 и 2). Каждое десятичное целое число имеет своё уникальное представление в каждой из этих систем. Это представление можно получить, последовательно деля число на 2 и 3 соответственно. Двоичная система является особенно важной, поскольку она оказалась наиболее удобной для компьютерной обработки данных.

В качестве примера головоломки, для решения которой используется преимущество двоичной системы, рассмотрим задачу из книги У. Паундстоуна [Pou03, p. 84].

Конверты с банкнотами. У вас есть тысяча банкнот по 1 доллару. Разложите их по 10 конвертам таким образом, чтобы можно

было выдать любую сумму от 1 до 1000 долларов, комбинируя эти конверты. Конечно же, выдача сдачи не предполагается.

Положим банкноты на сумму $1, 2, 2^2, \ldots, 2^8$ в первые девять конвертов, а оставшуюся сумму $1000 - (1+2+\cdots+2^8) = 489$ в десятый конверт. Любое число A, меньшее 489, может быть получено как комбинация степеней числа 2: $b_8 \cdot 2^8 + b_7 \cdot 2^7 + \cdots + b_0 \cdot 1$, где коэффициенты $b_8, b_7, \ldots, b_0$ равны 0 или 1. (Эти коэффициенты составляют представление числа A в двоичной системе. Самое большое число, которое можно представить с помощью девятизначного двоичного числа, это $2^8 + 2^7 + \cdots + 1 = 2^9 - 1 = 511$.) Любое число A от 489 до 1000 включительно можно представить как $489 + A'$, где $0 \leqslant A' \leqslant 511$. Таким образом, нужная сумма денег для такого A может быть получена как содержимое десятого конверта и комбинация первых девяти, которая задаётся двоичным представлением A'. Обратите внимание, что для некоторых A решение головоломки не единственное.

Хорошее упражнение для читателя — решение двух версий головоломки «*Задача Баше о гирях*» (№ 115), в которых используется двоичная и вариация троичной систем соответственно.

И наконец, многие задачи можно решить с помощью графов. *Граф* — это конечное множество точек на плоскости, некоторые из которых соединены линиями. Точки и линии называются *вершинами* и *рёбрами* графа соответственно. Рёбра могут не иметь ориентации или могут быть ориентированы от одной вершины к другой. В первом случае граф называется неориентированным, а во втором — ориентированным графом или кратко орграфом. В применении к головоломкам или играм вершины графа обычно представляют возможные состояния рассматриваемой задачи, а рёбра указывают на допустимые переходы между этими состояниями. Одна из вершин графа представляет исходное состояние, а другая — состояние, которое нужно достичь (таких вершин может быть несколько). Такой граф называется *графом пространства состояний*. Таким образом, представление задачи в виде графа сводит задачу к поиску пути между начальной вершиной и целевыми вершинами.

В качестве примера рассмотрим вариант очень старой и известной головоломки.[1)]

[1)]Классическая версия этой головоломки для трёх супружеских пар была включена в самый ранний из известных сборников математических задач на латыни *Propositiones ad Acuendos Juvenes*

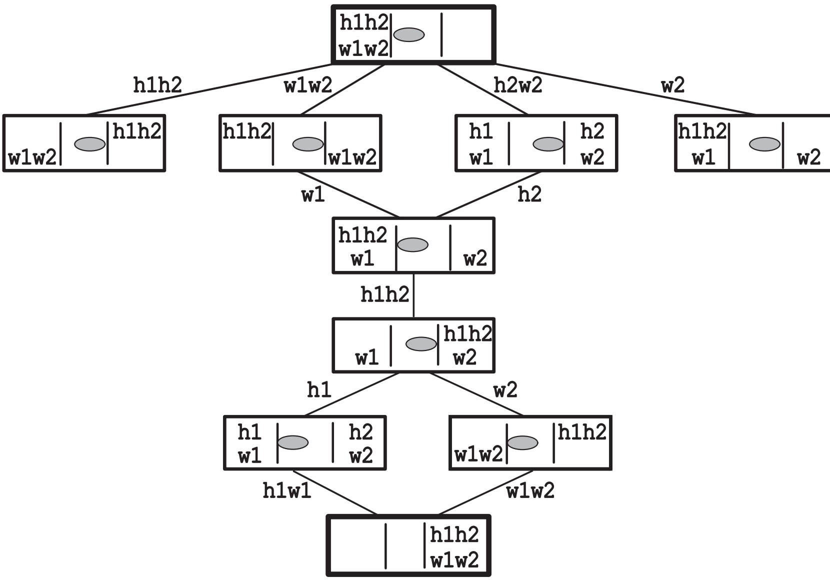


Рис. 1.5. Дерево пространства состояний для головоломки «*Два ревнивых мужа*»

Два ревнивых мужа. Двум супружеским парам нужно пересечь реку. У них есть лодка, в которую помещается не больше двух человек. К тому же оба мужа ревнивы и требуют, чтобы ни одна жена не находилась вместе с другим мужчиной в отсутствии мужа. Удастся ли им пересечь реку с такими ограничениями?

Граф пространства состояний для этой задачи показан на рис. 1.5. H_i, W_i обозначают мужа и жену в супружеской паре i ($i = 1, 2$) соответственно. Две полоски обозначают реку, а серый овал — местоположение лодки, которое определяет направление следующей поездки. (Для простоты на графе не показаны поездки через реку, которые отличаются очевидной заменой индексов; например, первыми пересекает реку первая пара H_1W_1 вместо второй H_2W_2.) Вершины графа, обозначающие исходное и конечное состояния, выделены жирной рамкой.

(«Задачи для заострения умов юношества»), составление которого приписывается Алкуину Йоркскому (ок. 735–804), известному средневековому учёному. Словесная формулировка задачи была немного грубовата по сегодняшним нормам [Had92].

На этом графе есть четыре кратчайших пути от вершины исходного состояния к вершине конечного состояния, каждый путь длиной пять рёбер. Вот эти пути, описанные рёбрами:

$$\begin{array}{ccccc} W_1W_2 & W_1 & H_1H_2 & H_1 & H_1W_1 \\ W_1W_2 & W_1 & H_1H_2 & W_2 & W_1W_2 \\ H_2W_2 & H_2 & H_1H_2 & H_1 & H_1W_1 \\ H_2W_2 & H_2 & H_1H_2 & W_2 & W_1W_2 \end{array}$$

Таким образом, существует четыре (не считая очевидных симметричных замен) оптимальных решения этой задачи, в каждом из которых реку нужно пересечь пять раз.

Головоломка «*Миссионеры и каннибалы*» (№ 49) является ещё одним примером задачи такого же рода.

Необходимо сделать два замечания по поводу решения головоломок с помощью представления в виде графа. Во-первых, для более сложных головоломок создание графа пространства состояний само по себе является алгоритмической задачей. Фактически задача может быть неосуществимой из-за очень большого числа состояний и преобразований. Например, у графа, представляющего состояния кубика Рубика, есть более чем 10^{19} вершин. Во-вторых, хотя конкретное расположение точек, представляющих вершины графа, не имеет теоретического значения, удачный выбор расположения рёбер на плоскости может помочь увидеть решение головоломки. Для примера рассмотрим следующую задачу, которую часто приписывают Паоло Гуарини (1512 г.), но на самом деле она была обнаружена в арабских манускриптах о шахматах, датируемых примерно 840 годом.

Головоломка Гуарини. На шахматной доске 3×3 расположены четыре коня: два белых коня стоят в двух нижних углах доски, а два чёрных — в двух верхних (рис. 1.6). Нужно поменять местами коней за минимальное число ходов так, чтобы белые кони оказались в верхних углах, а чёрные — в нижних.

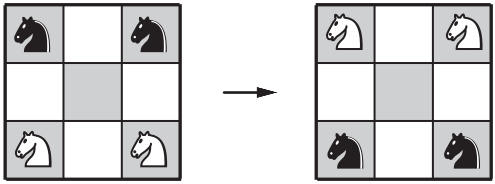
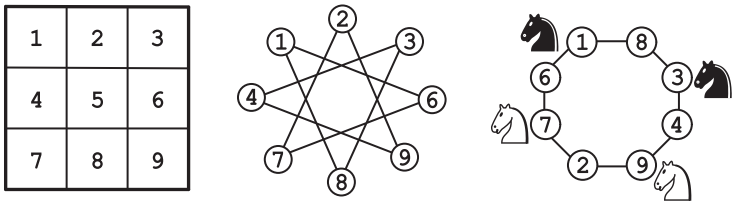


Рис. 1.6. «*Головоломка Гуарини*»



(*а*) (*б*) (*в*)

Рис. 1.7. (*а*) Нумерация клеток доски в *головоломке Гуарини.* (*б*) Простое представление графа головоломки. (*в*) Улучшенное представление графа головоломки

Представим клетки доски (пронумерованные для удобства на рис. 1.7,*а* последовательными цифрами) как вершины графа; рёбра графа соединяют две вершины, если конь может сделать ход между соответствующими клетками. Если мы расположим вершины, имитируя расположение клеток доски, мы получим граф, показанный на рис. 1.7,*б*. (Мы опускаем клетку номер 5 в центре доски, поскольку ни один из коней не может на неё попасть.) Однако видно, что граф на рис. 1.7,*б* не может помочь нам решить задачу. Если же мы разместим вершины вдоль окружности в том порядке, в котором на них попадает конь из вершины 1 (как показано на рис. 1.7,*в*), мы получим гораздо более понятную картину[1]. Из рис. 1.7,*в* ясно, что каждый допустимый ход любого коня сохраняет относительную расстановку коней в направлении по или против часовой стрелки. Таким образом, существует только два способа решить головоломку с помощью минимального количества ходов: передвигать коней вдоль рёбер в направлении по или против часовой стрелки до тех пор, пока каждый из коней не достигнет противоположного по диагонали края доски в первый раз. Для каждого из этих симметричных решений требуется 16 ходов.

В качестве упражнения по методу развёртывания графа мы рекомендуем решить головоломку «*Монеты на звезде*» (№ 34).

[1]Генри Дьюдени [Dud58, p. 230] назвал такое преобразование «методом пуговиц и верёвочек». Если представить вершины и рёбра графа как пуговицы и верёвочки соответственно, можно получить граф, показанный на рис. 1.7,*в*, путём поднимания и перетаскивания пуговиц 2, 8, 4 и 6 на противоположную сторону графа, чтобы «распутать» верёвочки.

Существуют также головоломки, которые можно решить, сводя их к математической задаче, такой как решение уравнения или нахождение максимума или минимума функции. Вот пример такой головоломки.

Как оптимально разделить пирог. На какое максимальное число кусков можно разрезать пирог прямоугольной формы, проводя n разрезов, если каждый разрез должен быть параллелен одной из сторон пирога, вертикальной или горизонтальной?

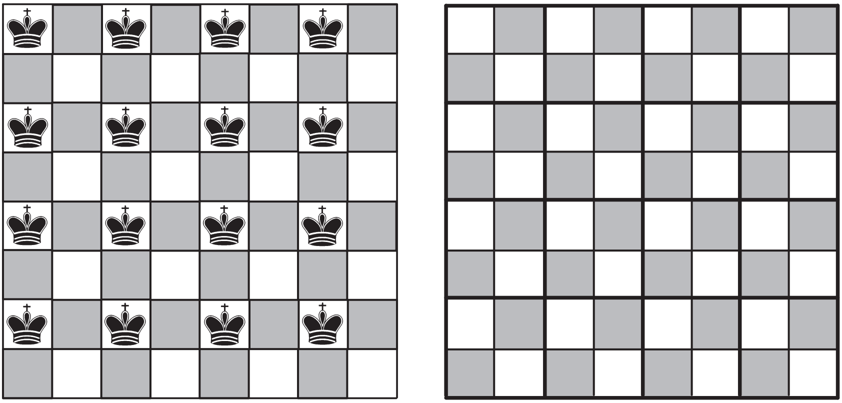
Если пирог разрезать h горизонтальными и v вертикальными разрезами, то общее количество кусков будет $(h + 1)(v + 1)$. Поскольку общее количество разрезов $h + v$ равно n, задача сводится к нахождению максимального значения функции

$$(h+1)(v+1) = hv + (h+v) + 1 = hv + n + 1 = h(n-h) + n + 1$$

для всех целочисленных значений h от 0 до n включительно. Поскольку $h(n-h)$ является квадратичной функцией от h, максимальное значение достигается при $h = \frac{n}{2}$, если n — чётное число и при $h = \frac{n}{2}$, округлённого в меньшую (обозначается $\left\lfloor \frac{n}{2} \right\rfloor$) или в большую сторону (обозначается $\left\lceil \frac{n}{2} \right\rceil$), если n является нечётным. Таким образом, головоломка имеет единственное решение $h = v = \frac{n}{2}$ для чётных n и два решения (которые можно считать симметричными) $h = \left\lfloor \frac{n}{2} \right\rfloor$, $v = \left\lceil \frac{n}{2} \right\rceil$ и $h = \left\lceil \frac{n}{2} \right\rceil$, $v = \left\lfloor \frac{n}{2} \right\rfloor$, если n нечётное.

Жадные алгоритмы

Так называемый «*жадный подход*» решает задачу оптимизации с помощью выбора последовательности шагов, на каждом из которых частичное решение расширяется до тех пор, пока не будет получено полное решение. На каждом шаге — и это является главной идеей стратегии — нужно получить максимальную немедленную выгоду, не нарушая при этом ограничения задачи. Такой жадный захват наилучшего варианта из возможных на каждом шаге делается в надежде, что последовательность локально оптимальных шагов в итоге приведёт к оптимальному решению всей задачи. Этот бесхитростный подход в некоторых случаях работает, а в некоторых — нет.



(*а*) (*б*)

Рис. 1.8. (*а*) Размещение 16 неатакующих королей. (*б*) Разделение доски для доказательства невозможности размещения более 16 неатакующих королей

Охотясь за головоломками, которые можно решить с помощью жадного подхода, не следует ожидать щедрой награды: хорошие головоломки обычно слишком «каверзные» для такого простого метода. Однако головоломки, которые можно решить с помощью жадного алгоритма, всё же существуют. В этих случаях, как правило, построить сам жадный алгоритм несложно, сложнее доказать, что он действительно даёт оптимальное решение. Приведём следующую головоломку в качестве примера.

Неатакующие короли. Расставьте на шахматной доске 8×8 наибольшее возможное количество королей так, чтобы никакие два короля не стояли на соседних клетках по вертикали, горизонтали или диагонали.

Следуя жадной стратегии, мы можем начать расставлять максимальное количество несоприкасающихся друг с другом королей (четырёх) на первую вертикаль доски. Затем, минуя вторую вертикаль, поскольку каждая из её клеток является соседней с одним из королей, ставим четырёх королей на третью вертикаль. Минуем четвёртую, и т. д., пока все 16 королей не окажутся на доске (рис. 1.8, *а*).

Чтобы показать, что расставить на доске более 16 неатакующих королей невозможно, разделим доску на 16 квадратов 2×2, как показано на рис. 1.8, *б*. Очевидно, что более одного короля в каждом квадрате разместить нельзя, т. е. общее количество неатакующих королей на доске не может превышать 16.

В качестве второго примера рассмотрим головоломку, которая стала особенно популярной после того, как её стали задавать во время собеседования при приёме на работу в *Microsoft*.

Переход по мосту ночью. Группе из четырёх человек, у которых есть один фонарик, нужно ночью пройти по шаткому мостику. Одновременно мостик могут перейти максимум два человека. У тех, кто переходит мостик (один человек или два), должен быть фонарик. Фонарик нужно переносить туда-сюда, перебросить его нельзя. Человек A переходит мост за 1 минуту, человек B — за 2 минуты, человек C — за 5 минут, а человек D — за 10. Когда два человека переходят вместе, они идут со скоростью наиболее медленного из них. Найдите кратчайшее время, за которое они все перейдут мост.

Жадный алгоритм, показанный на рис. 1.9, начинается с перехода на другую сторону двух самых быстроногих людей, т. е. A и B (это займёт две минуты). Затем A, как самый быстрый из двух, возвращается с фонариком (ещё 1 минута). Затем переходят два следующих быстрых человека A и C (5 минут), фонарик возвращается с быстрейшим A (1 минута). Последними переходят два оставшихся человека (10 минут). Всего по расчётам с использованием жадного алгоритма потребуется $(2+1)+(5+1)+10=19$ минут, но это *не* является оптимальным решением (см. эту же головоломку снова далее в книге (№ 7)).

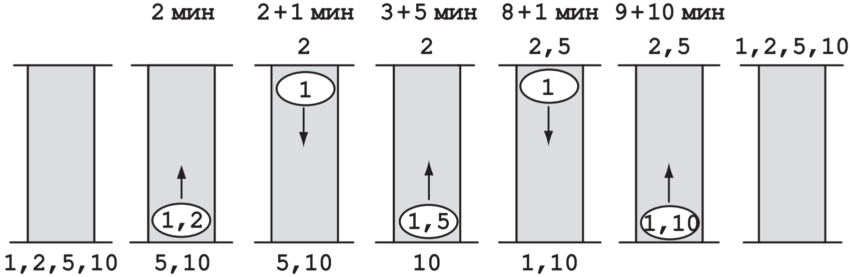


Рис. 1.9. Решение головоломки «*Переход по мосту ночью*» с использованием жадного алгоритма

Читателю будет полезно вернуться к головоломке «*Монеты на звезде*» (№ 34) и решить её с помощью жадного алгоритма, без тех преимуществ, которые даёт развёртывание графа.

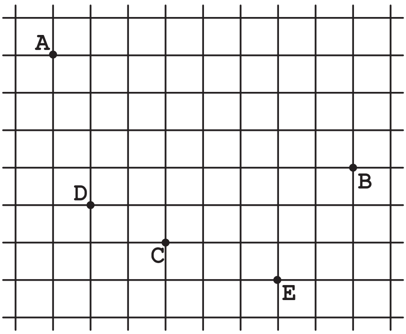
Итерационное улучшение

В то время как жадный алгоритм строит решение по частям, алгоритм итерационного улучшения начинает с некоторой легко получаемой аппроксимации решения и улучшает её путём повторения какого-либо простого шага. Чтобы установить состоятельность такого алгоритма, нужно убедиться, что алгоритм после конечного числа шагов останавливается и что полученная окончательная аппроксимация действительно является решением задачи. Рассмотрим следующую головоломку, которая является политически корректной версией задачи, описанной Мартином Гарднером в своей замечательной книге «*Есть идея!*» [Gar78, p. 131–132].

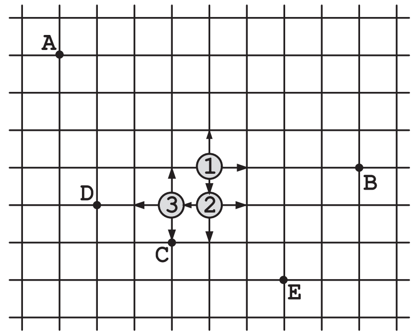
Где разместить киоск с лимонадом. Пятеро друзей — Алекс, Бренда, Кэти, Дэн и Эрл (Alex, Brenda, Cathy, Dan и Earl) — хотят установить киоск с лимонадом. Они живут в домах, обозначенных буквами A, B, C, D и E на рис. 1.10, *а*. В какой точке пересечения улиц они должны установить киоск, чтобы сумма расстояний от киоска до их домов была минимальной? Предполагается, что они измеряют расстояние от дома до киоска общим числом кварталов (по горизонтали и вертикали).

Первоначально друзья решили установить киоск в точке 1 (рис. 1.10, *б*), которая находится посередине по горизонтали между крайней левой точкой A и крайней правой точкой B и посередине по вертикали между самой верхней точкой A и самой нижней E. Но затем кто-то заметил, что это не самое лучшее из возможных расположений. И друзья решили применить такой итерационный алгоритм: рассмотреть по очереди места, находящиеся на расстоянии одного квартала от первоначальной точки в каком-нибудь порядке: например, сверху (к северу), справа (к востоку), снизу (к югу) и слева (к западу).

Как только новое место окажется ближе к их домам, заменяем старое место на новое и повторяем ту же самую операцию проверки. Если окажется, что ни одно из четырёх соседних мест не лучше, считаем последнее место оптимальным и останавливаем алгоритм. Действия алгоритма показаны на рис. 1.10, *б*, а вычисленные расстояния — на рис. 1.10, *в*.



(*а*)



(*б*)

	①	↑①	①→	①↓ = ②
A	4 + 3	4 + 2	5 + 3	4 + 4
B	4 + 0	4 + 1	3 + 0	4 + 1
C	1 + 2	1 + 3	2 + 2	1 + 1
D	3 + 1	3 + 2	4 + 1	3 + 0
E	2 + 3	2 + 4	1 + 3	2 + 2
Всего	23	26	24	22

	↑② = ①	②→	②↓	←② = ③
A		5 + 4	4 + 5	3 + 4
B		3 + 1	4 + 2	5 + 1
C		2 + 1	1 + 0	0 + 1
D		4 + 0	3 + 1	2 + 0
E		1 + 2	2 + 1	3 + 2
Всего		23	23	21

	↑③	③→ = ②	③↓	←③
A	3 + 3		3 + 5	2 + 4
B	5 + 0		5 + 2	6 + 1
C	0 + 2		0 + 0	1 + 1
D	2 + 1		2 + 1	1 + 0
E	3 + 3		3 + 1	4 + 2
Всего	22		22	22

(*в*)

Рис. 1.10. (*а*) Вариант головоломки «*Где разместить киоск с лимонадом*». (*б*) Шаги алгоритма. (*в*) Расстояния, вычисленные по алгоритму

Хотя похоже, что окончательное расположение, обозначенное на рис. 1.10,*б* цифрой 3, — это хороший выбор, алгоритм не даёт доказательства того, что этот выбор — наилучший. Другими словами, откуда мы узнаем, что не только четыре точки на расстоянии одного квартала от найденной точки являются худшим выбором, но и остальные тоже? Однако же, не будем волноваться о наших юных предпринимателях: это расположение действительно наилучшее, и у читателя есть возможность убедиться в этом, решив головоломку «*Выбор* *места*» (№ 74), которая является обобщением рассмотренной головоломки.

Вот ещё один пример головоломки, которую можно решить итерационным улучшением.

Положительные изменения. У нас есть таблица действительных чисел размером $m \times n$. Существует ли алгоритм, с помощью которого можно сделать суммы по всем строкам и по всем столбцам неотрицательными? Единственная операция, выполняемая алгоритмом, — изменение знаков всех чисел в любой строке или столбце.

Было бы естественным попытаться найти алгоритм, который на каждом шаге увеличивает число линий (строк и столбцов) с неотрицательными суммами. Однако, изменение знаков в какой-либо строке (столбце) с отрицательной суммой может сделать отрицательной сумму в каком-нибудь другом столбце (строке)! Остроумный способ преодолеть это затруднение — обратить внимание на общую сумму всех чисел в таблице. Поскольку она может быть вычислена или как сумма сумм по строкам, или по столбцам, изменение знаков в линии с отрицательной суммой наверняка увеличит общую сумму чисел в таблице. Таким образом, мы можем просто повторять поиск линии с отрицательной суммой. Если мы находим такую линию, то меняем знаки всех её чисел. Если не находим, значит, мы достигли цели и можем остановиться.

Это всё? Не совсем. Мы должны показать, что операции алгоритма не могут продолжаться без конца. И это действительно так, потому что при повторении операции алгоритма мы получаем конечное число различных таблиц (каждый из mn элементов может быть не более чем в двух положениях). Таким образом, количество сумм всех элементов тоже конечно. Поскольку алгоритм генерирует последовательность таблиц с увеличивающимися суммами, он должен становиться после конечного числа шагов.

В обоих примерах, приведённых выше, мы воспользовались некоторой величиной, обладающей следующими характеристиками:

- она изменяет своё значение только в нужном направлении (уменьшается в первой задаче и увеличивается во второй);
- она может принимать конечное число значений, что гарантирует остановку после конечного числа шагов;
- когда эта величина достигает финального значения, задача решена.

Такая величина называется *полуинвариантной*. Нахождение подходящей полуинвариантной величины может быть сложной задачей. Именно это делает головоломки с полуинвариантами популярными на математических турнирах. Например, вторая головоломка была использована в разделе практических задач на Первой Всероссийской математической олимпиаде в 1961 г. [Win04, p. 77]. Однако не стоит считать итерационное улучшение и полуинварианты математическими игрушками. На этом подходе основаны некоторые важные алгоритмы в информатике, например симплекс-метод. Заинтересованный читатель может найти несколько головоломок с полуинвариантами в разделе сложных головоломок данной книги.

Динамическое программирование

В информатике *динамическое программирование* применяется как метод решения задач с перекрывающимися подзадачами. Вместо того чтобы снова и снова решать перекрывающиеся подзадачи, динамическое программирование предлагает решить каждую из меньших подзадач только один раз и записать результаты в таблицу, из которой потом можно получить решение исходной задачи. Динамическое программирование было разработано известным американским математиком Ричардом Беллманом в 1950-х годах как общий метод оптимизации многостадийных процессов принятия решения. Для того чтобы задачу оптимизации можно было бы решить с помощью этого метода, задача должна обладать так называемой оптимальной подструктурой. В таком случае оптимальное решение может быть эффективно построено из оптимальных решений подзадач.

В качестве примера рассмотрим задачу подсчёта кратчайших путей.

Подсчёт кратчайших путей. Найдите число кратчайших путей из точки A в точку B. Точки находятся на пересечении улиц и проспектов. Улицы в городе строго горизонтальны, проспекты — строго вертикальны, как показано на рис. 1.11, *а*.

Пусть $P[i,j]$ — число кратчайших путей из точки A в точку пересечения улицы i $(1 \leqslant i \leqslant 4)$ и проспекта j $(1 \leqslant j \leqslant 5)$. Каждый кратчайший путь состоит из горизонтальных сегментов, идущих направо вдоль улиц, и вертикальных сегментов, идущих вниз вдоль проспектов. Таким образом, число кратчайших путей от A до пересечения i-й улицы и j-го проспекта может быть найдено как сумма числа кратчайших

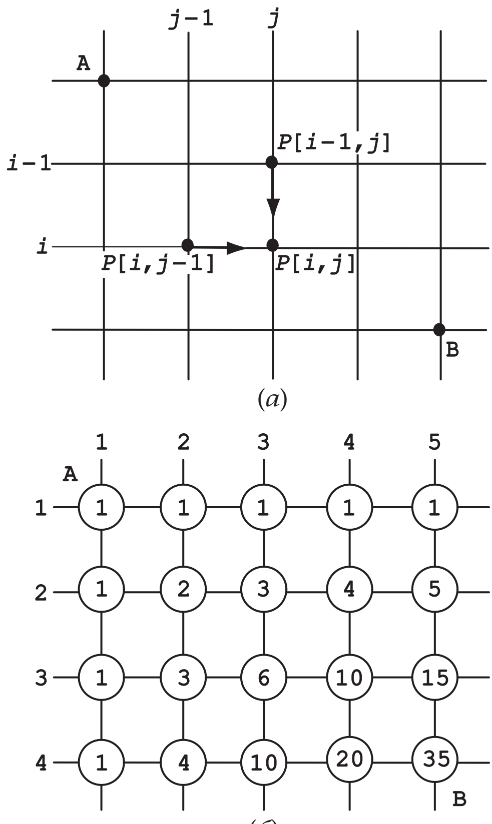


Рис. 1.11. (*а*) Использование динамического программирования для подсчёта числа кратчайших путей до пересечения (i,j). (*б*) Число кратчайших путей от А до каждого пересечения улиц и проспектов

путей от А до пересечения улицы $i-1$ и проспекта j (обозначенного как $P[i-1,j]$) и числа кратчайших путей от А до пересечения улицы i и проспекта $j-1$ (обозначенного как $P[i,j-1]$):

$$P[i,j] = P[i-1,j] + P[i,j-1] \quad \text{для любого } 1 < i \leqslant 4,\ 1 < j \leqslant 5,$$

где

$$P[1,j] = 1 \qquad \text{для любого } 1 \leqslant j \leqslant 5,$$
$$P[i,1] = 1 \qquad \text{для любого } 1 \leqslant i \leqslant 4.$$

Используя эти формулы, мы можем вычислить значения $P[i,j]$ или строка за строкой, начиная со строки 1 и двигаясь слева направо вдоль каждой строки, или же столбец за столбцом, начиная со столбца 1 и двигаясь вниз вдоль каждого столбца.

Эту задачу можно также решить с помощью простых комбинаторных рассуждений. Каждый кратчайший путь состоит из четырёх горизонтальных сегментов и трёх вертикальных. Пути отличаются друг от друга выбором трёх вертикальных сегментов из семи возможных. Таким образом, общее число кратчайших путей может быть получено как число способов выбрать три из семи, которое равняется[1)] $C(7,3) = \frac{7!}{3!4!} = 35$. Комбинаторный метод в этом простом примере приводит к решению быстрее, чем динамическое программирование. Однако для путей в менее регулярных сетках это зачастую не так. Чтобы в этом убедиться, предлагаем читателю решить головоломку «*Преграждённые пути*» (№ 13) из основного раздела книги.

Хотя использовать динамическое программирование зачастую весьма сложно, в качестве простого применения этой стратегии мы также советуем решить головоломки «*Спуск с максимальной суммой*» (№ 20) и «*Робот собирает монетки*» (№ 62).

МЕТОДЫ АНАЛИЗА АЛГОРИТМОВ

Хотя для многих головоломок в этой книге нужно просто построить алгоритм, некоторые из головоломок требуют анализа этого алгоритма. Цель этого раздела — дать обзор стандартных методов анализа алгоритмов, проиллюстрировав их на примере нескольких головоломок. Мы попытаемся сделать это на максимально доступном уровне; более углублённый обзор можно найти в таких учебниках, как [Lev06], [Kle05] и [Cor09], которые перечислены в порядке возрастания их сложности.

Анализ алгоритмов обычно служит для оценки их временнóй эффективности. Как правило, она определяется путём вычисления того, сколько раз нужно выполнить базовый шаг

[1)]Читатель, знакомый с элементарной комбинаторикой, заметит, что числа, стоящие на диагоналях, идущих с юго-запада на северо-восток на рис. 1.11, *б*, представляют собой элементы известного треугольника Паскаля (см., например, [Ros07, Section 5.4]).

алгоритма. Для большинства алгоритмов количество шагов растёт с увеличением размера задачи. Определить, *насколько быстро* оно возрастает, и является главной целью анализа алгоритмов. Для этого, что неудивительно, нужна математика. Поэтому начнём с обзора нескольких важных математических формул, которые чрезвычайно эффективны при анализе алгоритмов.

Несколько формул суммирования и эффективность алгоритмов

О Карле Фридрихе Гауссе (1777–1855), одном из величайших математиков, рассказывают такую известную историю. Когда Гауссу было около 10 лет, его учитель задал в классе задачу — просуммировать первые 100 положительных целых чисел

$$1 + 2 + \cdots + 99 + 100,$$

надеясь, что это займёт учеников на какое-то время. Учитель, конечно, не мог знать, что среди учеников присутствует математический гений. Карлу понадобилось лишь несколько минут, чтобы решить задачу с помощью группировки чисел по 50 парам, дающим одинаковую сумму 101:

$$(1 + 100) + (2 + 99) + \cdots + (50 + 51) = 101 \cdot 50 = 5050.$$

Обобщая этот приём на сумму первых n натуральных чисел, получаем формулу

$$1 + 2 + \cdots + (n - 1) + n = \frac{(n + 1)n}{2}. \tag{1}$$

В качестве упражнения мы рекомендуем читателю решить головоломку «*Счёт в уме*» (№ 9), которая использует данную формулу и её доказательство.

Формула (1) является практически незаменимой для анализа алгоритмов. Она также приводит к другим полезным формулам. Например, для суммы первых n положительных чётных чисел мы получаем

$$2 + 4 + \cdots + 2n = 2(1 + 2 + \cdots + n) = n(n + 1),$$

а для суммы первых n положительных нечётных чисел имеем

$$\begin{aligned} &1 + 3 + \cdots + (2n - 1) = \\ &= (1 + 2 + 3 + 4 + \cdots + (2n - 1) + 2n) - (2 + 4 + \cdots + 2n) = \\ &= \frac{2n(2n + 1)}{2} - n(n + 1) = n^2. \end{aligned}$$

Другая очень важная формула — сумма последовательных степеней числа 2; эту формулу мы уже использовали в первом разделе:

$$1 + 2 + 2^2 + \cdots + 2^n = 2^{n+1} - 1. \tag{2}$$

А теперь мы готовы рассмотреть наш первый пример анализа алгоритмов.

Изобретение шахмат. Скорее всего, игра в шахматы была изобретена много веков назад в северо-западной Индии мудрецом по имени Сисса. Когда Сисса показал своё изобретение царю, тому настолько понравилась эта игра, что он предложил изобретателю выбрать себе награду. Сисса попросил дать ему зерна из такого расчёта: одно-единственное зерно пшеницы нужно было положить на первую клетку шахматной доски, два — на вторую, четыре — на третью, восемь — на четвёртую, и т. д., пока все 64 клетки доски не будут заполнены. Была ли эта просьба со стороны изобретателя разумной?

В соответствии с формулой (2), общее количество зёрен, которое попросил Сисса, равняется

$$1 + 2 + \cdots + 2^{63} = 2^{64} - 1.$$

Если бы он считал со скоростью одно зерно в секунду, то общее количество времени для подсчёта всех зёрен равнялось бы примерно 585 миллиардов лет — более чем в 100 раз больше возраста планеты Земля. Это хорошая демонстрация невероятной скорости *экспоненциального роста*. Очевидно, что алгоритмы, которые требуют для своего решения времени, увеличивающегося экспоненциально с размером задачи, непрактичны почти во всех случаях, кроме очень маленьких задач.

Что было бы, если бы вместо удвоения количества зёрен на каждой клетке шахматной доски Сисса попросил добавлять по два зерна? Тогда общее количество зёрен было бы равно

$$1 + 3 + \cdots + (2 \cdot 64 - 1) = 64^2.$$

Если бы он считал с той же скоростью — одно зерно в секунду, — ему понадобилось бы менее 1 часа 14 минут, чтобы подсчитать свою заслуженную награду. Квадратичная скорость роста гораздо более приемлема для времени прогона алгоритма.

Ещё быстрее *линейные* алгоритмы. Для этих алгоритмов требуется время, пропорциональное размеру входных данных. Ещё эффективнее *логарифмические* алгоритмы. Эти алгоритмы обычно основываются на стратегии «уменьшение в постоянное число раз» (см. первую часть учебного раздела) и с каждым шагом уменьшают размер задачи, скажем, на половину. Это делает экспоненциальную скорость роста нашим союзником, быстро уменьшая ту часть задачи, которую ещё остаётся решить. Алгоритм игры «*Угадай число* (*двадцать вопросов*)», которую мы обсуждали в первой части учебного раздела, относится к этой категории.

Анализ нерекурсивных алгоритмов

Неудивительно, что под *нерекурсивным алгоритмом* подразумевается алгоритм, который не содержит рекурсий. Это означает, что он *не* работает, вызывая самого себя для решения всё более малых задач, до тех пор, пока задача не окажется элементарной с очевидным решением. Обычно нерекурсивный алгоритм можно анализировать, определив сумму числа выполнений основного шага. Затем эта сумма упрощается, чтобы найти или точную простую формулу для неё, или же приблизительную формулу, отражающую скорость роста. В качестве примера рассмотрим следующую задачу, напоминающую головоломку [Gar99, p. 88].

Построение квадратов. Алгоритм начинается с построения одного квадрата, и на каждой следующей итерации добавляются новые квадраты со всех внешних сторон. Сколько будет единичных квадратов после n-й итерации? Результаты первых нескольких итераций показаны на рис. 1.12.

Основной шаг данного алгоритма — добавление единичного квадрата. Следовательно, подсчёт основных шагов этого

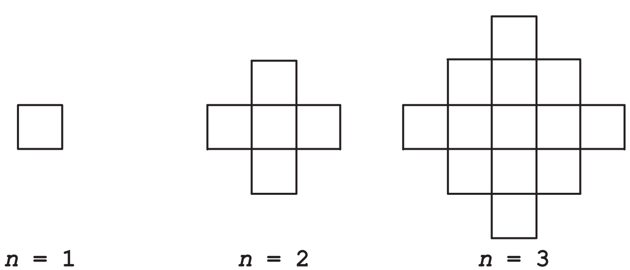


Рис. 1.12. Первые итерации алгоритма «*Построение квадратов*»

алгоритма просто сводится к подсчёту общего количества единичных квадратов. После n итераций в самом длинном горизонтальном ряду будет $(2n-1)$ таких квадратов. Ряды сверху и снизу будут содержать нечётное количество квадратов от 1 до $2n-3$. Поскольку сумма первых $n-1$ нечётных чисел равна $(n-1)^2$, общее количество единичных квадратов будет равно

$$2(1+3+\cdots+(2n-3))+(2n-1)=$$
$$=2(n-1)^2+(2n-1)=2n^2-2n+1.$$

Можно также решить эту головоломку, заметив, что число единичных квадратов, прибавляемых на i-й итерации $(1 < i \leqslant n)$, равно $4(i-1)$. Таким образом, общее число единичных квадратов после n итераций может быть получено как

$$1+4\cdot 1+4\cdot 2+\cdots+4(n-1)=1+4(1+2+\cdots+(n-1))=$$
$$=1+4\cdot\frac{(n-1)n}{2}=2n^2-2n+1.$$

Нет ничего плохого в том, чтобы использовать стандартные методы, но всегда стоит попытаться воспользоваться спецификой конкретной задачи. В данном случае можно было бы считать квадраты по диагоналям фигуры, полученной после n-й итерации. В этой фигуре n диагоналей с n единичными квадратами, которые чередуются с $n-1$ диагоналями с $n-1$ единичными квадратами, составляя в сумме $n^2+(n-1)^2=2n^2-2n+1$ квадратов.

В качестве ещё одного примера мы предлагаем читателю решить головоломку «*Подсчёт треугольников*» (№ 52).

Анализ рекурсивных алгоритмов

Мы проиллюстрируем стандартные методы анализа рекурсивных алгоритмов, рассмотрев классическую головоломку «Ханойская башня».

Ханойская башня. В обычном варианте этой головоломки имеются n колец различного размера и три стержня. Изначально все кольца находятся на первом стержне в порядке возрастания размера, самое большое — внизу, а самое маленькое — наверху. Необходимо последовательно переместить все кольца на другой стержень. За один раз можно переместить только одно кольцо, причём нельзя класть кольцо большего размера на меньшее.

У задачи есть элегантное рекурсивное решение, показанное на рис. 1.13. Для того чтобы переместить $n>1$ колец со

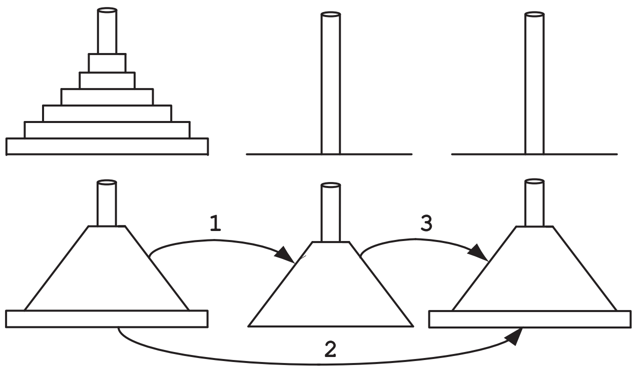


Рис. 1.13. Рекурсивное решение головоломки «*Ханойская башня*»

стержня 1 на стержень 3 (пользуясь стержнем 2 как вспомогательным), переместим $n - 1$ колец со стержня 1 на стержень 2 (пользуясь стержнем 3 как вспомогательным), затем переместим самое большое кольцо со стержня 1 на стержень 3 и, наконец, переместим $n - 1$ колец со стержня 2 на стержень 3 (используя стержень 1 как вспомогательный). Конечно, если $n = 1$, просто переместим это единственное кольцо непосредственно со стержня 1 на стержень 3.

Очевидно, что перемещение кольца с одного стержня на другой — это базовая операция алгоритма. Пусть $M(n)$ — это количество перемещений колец с помощью алгоритма, которое нужно выполнить для решения головоломки с n кольцами. Согласно описанию алгоритма (см. также рис. 1.13), мы получаем следующее соотношение для $M(n)$:

$$M(n) = M(n-1) + 1 + M(n-1) \quad \text{при } n > 1.$$

Естественно, его можно упростить:

$$M(n) = 2M(n-1) + 1 \quad \text{при } n > 1.$$

Такие соотношения называются *рекуррентными*, поскольку они описывают, как n-й член последовательности связан с предыдущими членами. В нашем случае соотношение показывает, что n-й член последовательности $M(n)$ на 1 больше, чем удвоенный предыдущий член $M(n-1)$. Обратите внимание, что последовательность определена не однозначно, поскольку ничего не говорится о её первом члене. Поскольку алгоритм предполагает всего одно перемещение для решения задачи с одним кольцом, мы имеет следующее дополнение к рекуррентному соотношению: $M(1) = 1$, которое, естественно, называется *начальным условием*. В итоге мы имеем следующее рекуррентное соотношение с начальным условием, описывающее количество перемещений с помощью рекурсивного алгоритма для решения головоломки «Ханойская башня» с n кольцами:

$$M(n) = 2M(n-1) + 1 \quad \text{при } n > 1,$$
$$M(1) = 1.$$

Вместо того, чтобы использовать какой-либо стандартный метод для решения таких уравнений, который можно найти в учебниках, упомянутых в первой части учебного раздела, попробуем применить индуктивный подход: вычислим первые несколько значений $M(n)$ с помощью вышеупомянутых формул, затем попробуем определить общую схему, а потом доказать, что эта схема верна для всех положительных n.

n	$M(n)$
1	1
2	3
3	7
4	15

Обзор первых значений $M(n)$ позволяет предположить, что формулой является $M(n) = 2^n - 1$. Очевидно, $M(1) = 2^1 - 1 = 1$. Самый лёгкий путь доказать, что это верно для всех $n > 1$,— это подставить формулу в уравнение, чтобы посмотреть, получим ли мы равные значения для всех таких n. Это и в самом деле так, поскольку

$$M(n) = 2^n - 1 \quad \text{и} \quad 2M(n-1) + 1 = 2(2^{n-1} - 1) + 1 = 2^n - 1.$$

Таким образом, мы имеем экспоненциальный алгоритм, который даже при средних значениях n будет работать в течение невероятно долгого времени. И это вовсе не потому, что этот конкретный алгоритм плох; на самом деле нетрудно доказать, что для данной задачи он является наиболее эффективным из всех возможных алгоритмов. Всё дело в сложности, присущей данной задаче, которая делает вычисления столь долгими. Но есть и хорошие новости: в оригинальной версии головоломки «Ханойская башня», которая была опубликована её изобретателем, французским математиком Эдуардом Люка́ в 1883 г., сказано, что наш мир закончит своё существование, когда 64 кольца будут перемещены монахами мистической башни Брахмы. Исходя из предположения, что монахи не едят, не спят, не умирают и перемещают одно кольцо за минуту, конец нашему миру придёт примерно через $3 \cdot 10^{13}$ лет, что более чем в тысячу раз больше, чем предполагаемый возраст нашей Вселенной.

В качестве упражнения можно решить рекуррентное соотношение минимального количества перемещений в головоломке «*Ханойская башня с ограничением*» (№ 83) — одном из многочисленных вариантов классической версии:

$$M(n) = 3M(n-1) + 2 \quad \text{для } n > 1,$$
$$M(1) = 2.$$

Инварианты

В конце учебного раздела кратко расскажем о понятии инварианта. В нашем случае *инвариант* — это свойство, которое остаётся неизменным при решении задачи с помощью алгоритма. При решении головоломок инвариант часто используют, чтобы показать, что задача не имеет решения, поскольку инвариантное свойство выполняется на начальном этапе решения головоломки и не выполняется на финальной стадии. Рассмотрим несколько примеров.

Покрытие фигурками домино шахматных досок с дефектами.

а) Можно ли замостить фигурками домино шахматную доску 8×8, у которой нет одной угловой клетки (рис. 1.14, *а*)?

б) Можно ли замостить фигурками домино шахматную доску 8×8, у которой нет двух угловых клеток, противоположных по диагонали (рис. 1.14, *б*)?

Ответ на первый вопрос «нет». При любом покрытии фигурками домино количество покрытых клеток всегда чётное (чётность числа покрытых клеток и является здесь инвариантом), а число клеток на данной доске — нечётное.

Ответ на второй вопрос тоже «нет», хотя число клеток на доске очевидно чётное. В этой задаче инвариант другой: поскольку одна костяшка домино покрывает одну тёмную и одну светлую клетку, в итоге количество покрытых тёмных и светлых клеток оказывается всегда одинаковым. В данной задаче число тёмных и светлых клеток на доске отличается на два, поэтому замостить такую доску невозможно.

Вообще, контроль чётности или нечётности и раскраска — это две наиболее широко используемые идеи применения понятия инварианта. Мы советуем читателю решить

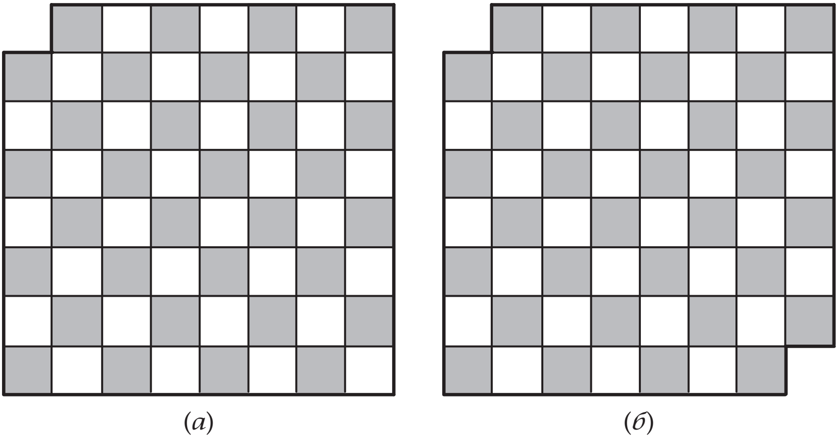


Рис. 1.14. (*а*) Шахматная доска, у которой нет одной угловой клетки. (*б*) Шахматная доска, у которой нет двух клеток в углах, противоположных по диагонали

типичные головоломки этого рода «*Последний шар*» (№ 50) и «*Проход из угла в угол шахматной доски*» (№ 18).

Важность понятия инварианта в различных ситуациях становится очевидной, если вспомнить известную головоломку о прогулках по старому прусскому городу Кёнигсбергу.

Задача о кёнигсбергских мостах. Можно ли во время одной прогулки пройти по всем семи мостам Кёнигсберга только по одному разу и вернуться в начальную точку? Схематический план реки с двумя островами и семью мостами показан на рис. 1.15.

Эта головоломка была решена великим математиком швейцарского происхождения Леонардом Эйлером (1707–1783). Для начала Эйлер понял, что путь по суше (по

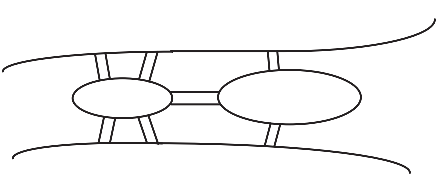


Рис. 1.15. Схематический план семи мостов через реку в Кёнигсберге, соединяющих основную часть города и два острова

берегу реки или по острову) не влияет на решение задачи. Единственная существенная информация — это расположение мостов. В современных терминах, эта догадка позволила ему преобразовать данную задачу в задачу о графе, как показано на рис. 1.16. (Фактически это *мультиграф*, поскольку некоторые из его вершин соединены более чем одним ребром.)

Тогда вопрос сводится к тому, содержит ли мультиграф на рис. 1.16 *эйлеров цикл*, т. е. последовательность смежных вершин графа, по которым можно пройти по рёбрам графа и вернуться в начальную вершину, причём пройдя по каждому ребру ровно один раз. Эйлер обратил внимание, что такой цикл должен входить в вершину столько же раз, сколько и выходить из неё. Таким образом, эйлеров цикл может существовать только в мультиграфе, в котором число рёбер, примыкающих к вершине (оно называется *степенью* вершины) — чётное для каждой вершины.

Это инвариантное свойство показывает, что у задачи о кёнигсбергских мостах нет решения, поскольку это необходимое условие не удовлетворяется: у каждой вершины на мультиграфе, показанном на рис. 1.16, нечётная степень. Более того, анализ показывает, что невозможно пройти по всем мостам по одному разу, даже если не возвращаться в исходную точку. Такой путь называется *эйлеровым путём*. Для его существования необходимо, чтобы все вершины мультиграфа имели чётную степень, за исключением двух вершин, в которых путь начинается и заканчивается.

Между прочим, эти условия оказываются не только необходимыми, но и достаточными для существования эйлерова цикла и эйлерова пути в связном мультиграфе. (Мультиграф называется *связным*, если есть путь между каждой парой его вершин. Естественно, в противном случае

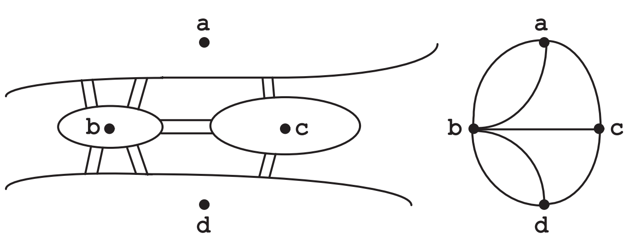


Рис. 1.16. Мультиграф задачи о кёнигсбергских мостах

не может существовать ни эйлеров цикл, ни эйлеров путь.) Этот факт был замечен самим Эйлером и затем формально доказан другим математиком. Читатель может воспользоваться этим анализом для решения головоломки «*Обвести фигуру*» (№ 28) из основного раздела книги.

Сегодня задача о кёнигсбергских мостах считается основополагающей в теории графов, важном разделе математики с приложениями в информатике и исследовании операций.

Эту задачу также часто приводят как пример, чтобы показать потенциальную пользу головоломок для серьёзной науки, образования и практических применений.

В следующей головоломке инвариант используется не для того, чтобы доказать, что решения не существует.

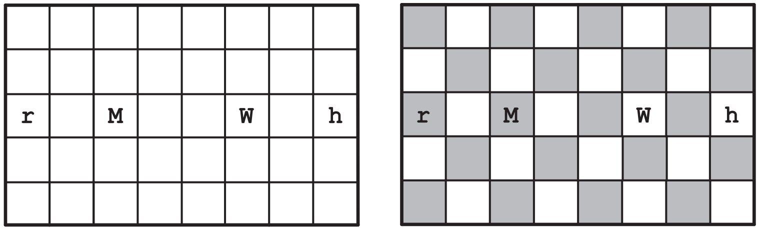
Разделить плитку шоколада. Нужно разделить плитку шоколада размером $n \times m$ на nm квадратиков, разломав плитку минимальное количество раз. Всё меньшие плитки можно делить только по прямой линии и можно разламывать только одну плитку за один раз.

Эта головоломка хорошо известна среди математиков и специалистов по информатике. Чтобы оценить её прелесть, читателю следует прервать чтение и попытаться её решить самому. Объяснение решения даётся в двух следующих предложениях. За один раз можно разламывать только один кусочек, поэтому каждый раздел увеличивает количество кусочков на 1. Таким образом, чтобы получить из одной плитки размером $n \times m$ всего nm кусочков размером 1×1, нужно разломать плитку $nm - 1$ раз. Это можно сделать, ломая плитку в *любой* последовательности $nm - 1$ раз.

В заключение рассмотрим головоломку, где инвариант играет более конструктивную роль, подсказывая способ, как построить алгоритм. Эту головоломку опубликовали два самых известных создателя головоломок в истории — Генри Э. Дьюдени [Dud02, p. 95] и Сэм Лойд [Loy59, p. 8] — в различных вариантах (разный размер доски и формулировка).[1)]

Цыплята в огороде. Поле размером 5×8 представляет собой огород с кукурузными побегами. Две фигурки одного цвета — фермер и его жена, две фигурки другого цвета — петушок и курочка. За каждый ход человек или цыплёнок перемещаются на один квадратик вверх, вниз, направо или налево, но не по диагонали. На рис. 1.17, *а*

[1)] Дьюдени и Лойд в течение нескольких лет сотрудничали. Затем Дьюдени прервал переписку, обвинив Лойда в воровстве головоломок и публикации их под своим именем.



(*а*) (*б*)

Рис. 1.17. (*а*) Доска с условиями головоломки «*Цыплята в огороде*». (*б*) Доска, раскрашенная как шахматная

показана начальная позиция. Мужчина (M) и женщина (W) каждый делают по одному ходу, затем петушок (r) и курочка (h) тоже делают по одному ходу. Игра продолжается, пока цыплята не будут пойманы. Цыплёнок считается пойманным, когда фермер или его жена переходят на клетку, на которой находится цыплёнок. Нужно поймать цыплят за минимальное число ходов.

Раскрасим доску как шахматную (рис. 1.17, *б*). Тогда легко понять, что мужчина не сможет поймать петушка, а женщина — курочку. В самом деле, человек сможет поймать цыплёнка только в том случае, когда они находятся на двух соседних клетках, а две соседние клетки всегда разного цвета. Но мужчина и петушок начинают игру на одинаково окрашенных клетках, и это свойство сохраняется после того, как они сделают любое конечное число ходов. Это же наблюдение относится и к женщине с курочкой. Поэтому мужчина должен гоняться за курочкой, а женщина — за петушком. Даже если цыплята пытаются убежать не согласованно друг с другом, один из них будет пойман на восьмом ходу, а другой — на девятом.

На этом мы заканчиваем учебный раздел. Что касается вопроса, какую стратегию нужно применять для той или иной головоломки, ответа на него нет. (Если бы он существовал, решение головоломок потеряло бы привлекательность в качестве интеллектуального развлечения.) Различные стратегии — это просто инструменты, которые могут годиться или нет для конкретной головоломки. Однако, практикуясь, человек может развить некоторое интуитивное понимание, какая стратегия подойдёт. Но такая интуиция, конечно же, работает далеко не всегда.

Всё же стратегии и методы, которые мы обсудили, составляют очень полезный набор инструментов для решения алгоритмических задач. Они более конкретны, чем аналогичные инструменты, находящиеся в арсенале математиков (несмотря на попытки их конкретизировать, сделанные, например, в знаменитой книге Д. Пойа «Как решать задачу» [Pol57]).

Конечно, даже если человек знает, какую стратегию следует применить, её применение может быть далеко не простым делом. Например, чтобы доказать, что у головоломки нет решения, обычно применяется инвариант. Но найти подходящий инвариант для конкретной задачи бывает нелегко, даже если попытаться использовать чётность или раскраску доски. Повторим, что с практикой решать задачи становится легче, но всё же не всегда совсем легко.

Глава 2

ГОЛОВОЛОМКИ

ЛЁГКИЕ ГОЛОВОЛОМКИ

1. Волк, коза и капуста

Крестьянин находится на берегу реки с волком, козой и кочаном капусты. Ему нужно перевезти всех троих на другой берег реки в лодке. Однако места в лодке хватит только на двоих — на самого крестьянина и ещё на одного (или на волка, или на козу, или на капусту). В отсутствии крестьянина волк может съесть козу, а коза — капусту. Подскажите крестьянину, как ему перевезти всех трёх «пассажиров» на другой берег.

2. Выбор перчаток

В ящике хранится 20 перчаток: 5 пар чёрных, 3 пары коричневых и 2 пары серых. Вы выбираете перчатки в темноте и можете посмотреть, что вы выбрали, только после того, как выбор сделан. Чему равно минимальное количество перчаток, которые надо взять из ящика, чтобы гарантировано получить:

а) по крайней мере одну подходящую пару;

б) по крайней мере одну подходящую пару каждого цвета?

3. Разделение прямоугольника

Найдите все значения $n > 1$, при которых можно разделить прямоугольник на n прямоугольных треугольников и разработайте алгоритм для этого.

4. Отряд солдат

Отряду из 25 солдат нужно переправиться через широкую и глубокую реку. Моста по близости не видно. Они замечают двух 12-летних мальчиков, играющих в гребной шлюпке. Шлюпка небольшая, может вместить только двух мальчиков или одного солдата. Как солдаты смогут перебраться через реку? Обоих мальчиков в конце надо оставить в шлюпке на берегу. Сколько раз в этом алгоритме шлюпка должна пройти от берега к берегу?

5. Перестановки строк и столбцов

Можно ли преобразовать таблицу слева в таблицу справа на рис. 2.1 с помощью перестановок строк и перестановок столбцов?

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→

12	10	11	9
16	14	5	13
8	6	7	15
4	2	3	1

Рис. 2.1. Исходная и конечная таблицы в головоломке «*Перестановки строк и столбцов*»

6. Счёт на пальцах

Маленькая девочка считает от 1 до 1000 по пальцам левой руки в следующем порядке. Большой палец — 1, указательный — 2, средний — 3, безымянный — 4, мизинец — 5. Затем она меняет направление: безымянный — 6, средний — 7, указательный — 8 и большой — 9. После этого указательный — 10 и т. д. Если она будет считать таким образом, на каком пальце она остановит счёт?

7. Переход по мосту ночью

Четыре человека хотят перейти шаткий мостик ночью. Все они вначале находятся на одной стороне реки. Темно, у них есть один фонарик. Одновременно мостик могут перейти максимум два человека. У тех, кто переходит мостик (один человек или два) должен быть с собой фонарик. Фонарик нужно переносить туда и обратно, перебросить его нельзя. Первый человек переходит мост за 1 минуту, второй — за 2 минуты, третий — за 5 минут, а четвёртый — за 10. Когда два человека идут вместе, они идут со скоростью наиболее медленного из них. Например, если первый идёт вместе с четвёртым, то у них проход по мосту занимает 10 минут. Если четвёртый возвращается с фонариком, то это занимает уже 20 минут. Смогут ли они все перейти мост за 17 минут?

8. Как собрать пазл?

В пазле 500 кусочков. «Секция» пазла — это набор из одного или более кусочков, которые соединены друг с другом. «Ход» — это соединение двух секций. За какое минимальное количество ходов можно собрать пазл?

9. Счёт в уме

Таблица 10×10 заполнена повторяющимися числами по диагоналям, как показано на рис. 2.2. Подсчитайте в уме сумму всех чисел в таблице.

1	2	3			…			9	10
2	3						9	10	11
3						9	10	11	
					9	10	11		
				9	10	11			
⋮			9	10	11				⋮
		9	10	11					
	9	10	11						17
9	10	11						17	18
10	11				…		17	18	19

Рис. 2.2. Таблица чисел, общую сумму которых нужно найти в головоломке «*Счёт в уме*»

10. Фальшивая монета из восьми

У нас есть восемь монет, которые выглядят одинаково. Известно, что одна из монет — фальшивая и что она легче настоящих. За какое минимальное число взвешиваний на чашечных весах без гирь можно определить фальшивую монету?

11. Столбик фальшивых монет

Есть 10 столбиков по 10 монет. Все монеты выглядят одинаково. В одном из столбиков все монеты фальшивые, а во всех других — настоящие. Каждая настоящая монета весит 10 грамм, а каждая фальшивая — 11 грамм. У вас есть цифровые весы, которые показывают точный вес любого количества монет. За какое минимальное количество взвешиваний можно определить столбик с фальшивыми монетами?

12. Можно ли замостить доску?

Можно ли замостить доску размером 8×8 фигурками домино (плиточками размером 2×1) так, чтобы никакие два домино не образовали квадрат 2×2?

13. Преграждённые пути

Найдите количество различных кратчайших путей из точки A в точку B в городе с идеально горизонтальными улицами и вертикальными проспектами, как показано на рис. 2.3. Ни

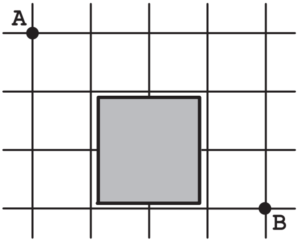


Рис. 2.3. Сеть улиц города с огороженным участком (обозначенным серым цветом)

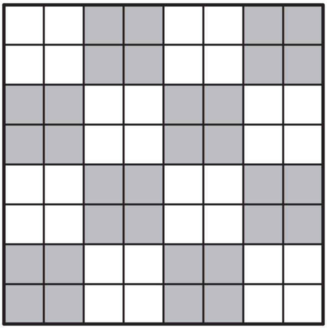


Рис. 2.4. Доска размером 8×8, которую нужно переделать в обычную шахматную доску

один путь не должен проходить через ограждённый забором участок, обозначенный на рисунке серым цветом.

14. Переделать шахматную доску

На шахматной доске размером 8×8 квадраты были ошибочно покрашены двумя цветами так, как показано на рис. 2.4. Разрежьте эту доску по горизонтальным и вертикальным линиям так, чтобы можно было из частей собрать обычную шахматную доску размером 8×8. Каково минимальное количество частей, на которые нужно разрезать доску? Как их нужно потом собрать?

15. Замостить доску плитками тримино

Для каждого из перечисленных ниже случаев докажите или опровергните утверждение, что при всех $n > 0$ доску можно замостить угловыми тримино. Размеры досок следующие:

а) $3^n \times 3^n$; б) $5^n \times 5^n$; в) $6^n \times 6^n$.

Напомним, что угловое тримино — это многоугольник L-образной формы, составленный из трёх примыкающих друг к другу квадратов (см. первую часть учебного раздела). При покрытии доски тримино можно ориентировать различными способами, но без наложения оных друг на друга. Необходимо, чтобы были покрыты все квадраты доски.

16. Печём блины

Вам нужно испечь $n \geqslant 1$ блинов на сковородке, на которой помещается одновременно только два блина. Каждый блин нужно обжарить с обеих сторон. Одна сторона жарится за одну минуту, независимо от того, один или два блина на сковородке. Придумайте алгоритм, как выполнить работу за минимальное время. Выразите это минимальное время как функцию от n.

17. Куда дойдёт король?

а) Шахматный король может ходить на любое соседнее поле по горизонтали, вертикали или диагонали. Предположим, что король вначале стоит на каком-либо поле бесконечной шахматной доски. На скольких различных полях он может находиться через n ходов?

б) Ответьте на этот же вопрос, если король не может ходить по диагонали.

18. Проход из угла в угол шахматной доски

Сможет ли конь, изначально стоящий в нижнем левом углу стандартной шахматной доски 8×8, побывать на всех полях в точности по одному разу и прийти в верхний правый угол? (Ход коня — это прыжки буквой «Г»: на две клетки по горизонтали или вертикали, затем на одну клетку в перпендикулярном направлении.)

19. Нумерация страниц

Страницы в книге пронумерованы последовательно, начиная с 1. Всего в номерах страниц 1578 знаков. Сколько страниц в книге?

20. Спуск с максимальной суммой

Несколько натуральных чисел выстроены в треугольник, как на рис. 2.5. Разработать алгоритм (конечно, более эффективный, чем полный перебор) для поиска наибольшей возможной суммы при спуске от вершины треугольника к его основанию через последовательность соседних чисел (по одному числу на каждый уровень).

2
5 4
3 4 7
1 6 9 6

Рис. 2.5. Треугольник из чисел, в котором путь, соответствующий наибольшей возможной сумме, обозначен кружками

21. Разбиение квадрата

Найдите все значения n, при которых можно разбить квадрат на n более мелких квадратов, и разработайте алгоритм для этого разбиения.

22. Упорядочение списка команд

Предположим, что вы знаете результаты окончившегося кругового турнира, в котором каждая из n команд сыграла с другой по одному разу. Если известно, что ничейных результатов не было, всегда ли можно составить список команд таким образом, чтобы каждая команда выиграла игру с командой, находящейся в списке непосредственно после неё?

23. Задача о польском национальном флаге

На столе лежит ряд из $n > 1$ шашек. Некоторые из них — красные, а некоторые — белые. (Белый и красный — цвета национального флага Польши.) Придумайте алгоритм для перестановки шашек таким образом, чтобы сначала шли все красные, а затем — все белые. Алгоритм допускает только две операции: определение цвета шашки и перестановка двух шашек. Попытайтесь минимизировать количество перестановок.

24. Раскрашивание шахматной доски

Для каждой из шахматных фигур найдите минимальное количество цветов, в которые нужно раскрасить шахматную доску размером $n \times n$ ($n > 1$), чтобы никакие две фигуры, если их поставить на клетки одинакового цвета, не угрожали друг другу.

а) Конь. (Конь атакует любую клетку, находящуюся от него в двух клетках по горизонтали и одной по вертикали, или в двух клетках по вертикали и одной по горизонтали.)

б) Слон. (Слон атакует любую клетку, находящуюся с ним на одной диагонали.)

в) Король. (Король атакует любую соседнюю с ним клетку по горизонтали, вертикали или диагонали.)

г) Ладья. (Ладья атакует любую клетку, находящуюся с ней на одной горизонтали или вертикали.)

Клетки, которые могут атаковать эти фигуры, показаны на рис. 2.6.

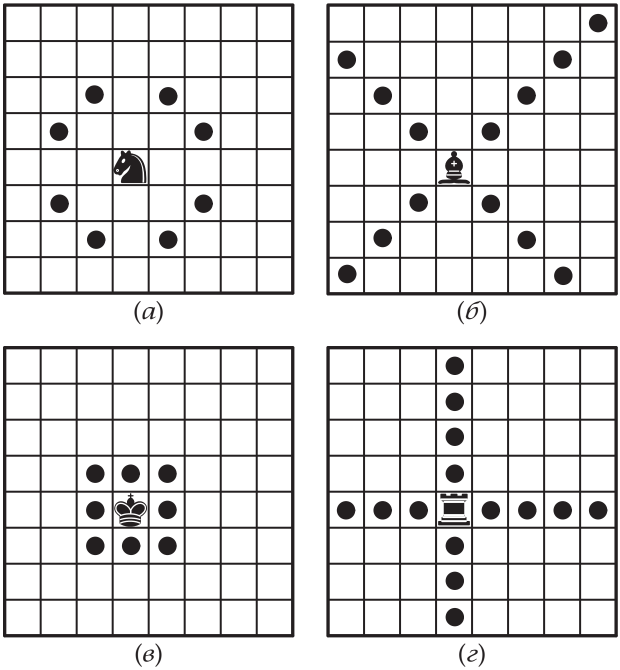


Рис. 2.6. Клетки, которым угрожает
(*а*) конь, (*б*) слон, (*в*) король, (*г*) ладья

25. Лучшее время для жизни

Редактор книги «История мировой науки» хочет узнать, в какое время на Земле жило наибольшее число выдающихся учёных. Выдающийся учёный, по определению, это человек, чьё имя записано в этой книге вместе с датами его рождения и смерти (ныне живущих учёных в книге нет). Придумайте алгоритм, который поможет редактору добиться цели. Входными данными для алгоритма должен служить именной указатель книги, в котором фамилии идут в алфавитном порядке и сопровождаются годами рождения и смерти учёных. Если учёный A умер в том же году, в котором родился B, предполагаем, что первое произошло раньше, чем второе.

26. Тьюринг в списке

Если составить список всех возможных «слов» из букв G, I, N, R, T и U, упорядоченных в алфавитном порядке, начиная с GINRTU и заканчивая UTRNIG, то на каком месте в таком списке будет TURING? (Алан Тьюринг (1912–1954) — английский математик и специалист по информатике, который, помимо прочих выдающихся достижений, сыграл главную роль в создании теоретической информатики.)

27. Игра «Икосиан»

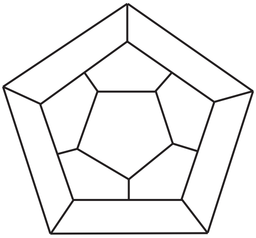
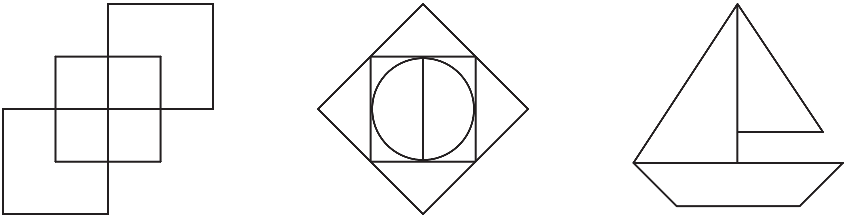


Рис. 2.7. Граф для головоломки «*Игра „Икосиан"*»

В XIX веке известный ирландский математик Сэр Уильям Гамильтон (1805–1865) изобрёл головоломку, которая была представлена миру как «*Игра „Икосиан"*». В неё играли на деревянной доске с отверстиями, обозначающими самые крупные города мира, и желобками, означающими пути между ними (см. рис. 2.7). Целью игры было найти обходной путь, проходящий через все города ровно по одному разу и возвращающийся в начальную точку. Сможете ли вы найти такой путь?

28. Обвести фигуру

Для каждой из трёх фигур на рис. 2.8: либо обведите фигуру, не отрывая ручки от бумаги и не возвращаясь ни по одной из линий, либо докажите, что это невозможно.



(*а*) (*б*) (*в*)

Рис. 2.8. Попробуйте обвести три фигуры

29. Ещё раз о магическом квадрате

Магический квадрат порядка три — это таблица 3×3, заполненная девятью различными числами от 1 до 9 так, что сумма чисел в каждой строке, каждом столбце и по обеим диагоналям одна и та же. Найдите *все* магические квадраты порядка 3.

30. Ломание палки

Палку длиной в 100 единиц нужно разломить на 100 единичных частей. Какое для этого требуется минимальное число разломов, если можно ломать несколько частей палки одновременно? Также вкратце изложите алгоритм, который позволяет сделать то же с палкой длиной в n единиц с минимальным числом разломов.

31. Трюк с тремя стопками карт

Фокусник просит кого-нибудь выбрать одну из 27 карт из колоды и вернуть её обратно, не показывая фокуснику. После этого фокусник тасует колоду и раскладывает все карты на три стопки лицом кверху, по одной карте в каждую стопку за раз. У человека, который загадал карту, фокусник спрашивает, в какой стопке она находится. Потом фокусник помещает стопку с этой картой в колоду между двух других и, не перетасовывая, распределяет карты на три стопки, как раньше. После того как фокусник ещё раз спрашивает, в какой стопке выбранная карта, он помещает эту стопку между двумя другими и в последний раз распределяет карты на три стопки. Когда ему вновь говорят, в какой стопке выбранная карта, он называет эту карту. Объясните фокус.

32. Турнир на выбывание

В отборочных соревнованиях — таких, как теннисные турниры Большого шлема, — каждый проигравший игрок немедленно выбывает из последующих раундов игры. В итоге остаётся единственный победитель. Если в начале турнира участвует n игроков, определите:

а) Сколько нужно сыграть матчей, чтобы определить победителя?

б) Сколько раундов в таком турнире?

в) Сколько ещё нужно матчей, чтобы определить игрока, занявшего второе место?

33. Магия и псевдомагия

а) У нас есть таблица $n \times n$, ячейки которой нужно заполнить числами от 1 до 9 включительно, по одному числу в ячейке. Сделайте это таким образом, чтобы каждый квадрат размером 3×3 внутри таблицы представлял собой магический квадрат. Найдите все значения $n \geqslant 3$, для которых это возможно.

б) Найдите все значения $n \geqslant 3$, для которых можно заполнить таблицу $n \times n$ числами от 1 до 9 включительно таким образом, чтобы каждый квадрат размером 3×3 являлся полумагическим квадратом. В *полумагическом квадрате* суммы по всем строкам и столбцам должны быть одинаковые, а по диагоналям могут различаться.

34. Монеты на звезде

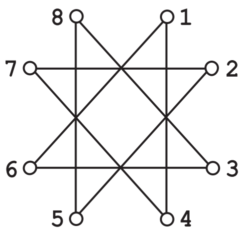


Рис. 2.9. Звезда, на концах которой нужно разместить монеты

Цель головоломки — разместить максимально возможное количество монет на концах восьмиконечной звезды, изображённой на рис. 2.9. Монеты нужно класть одну за другой следующим образом: 1) сначала монету нужно положить на свободный конец, а затем сдвинуть вдоль какой-нибудь линии на другой свободный конец; 2) как только монета размещена таким образом, переместить её снова нельзя. Например, можно начать, поместив первую монету на точку 6 и сдвинув её на точку 1 (обозначим $6 \to 1$), где монета и должна оставаться. Можно продолжить, например, так: $7 \to 2$, $8 \to 3$, $7 \to 4$, $8 \to 5$. Таким образом мы разместим пять монет.

35. Три кувшина

У вас есть 8-литровый кувшин, наполненный водой, и два пустых кувшина ёмкостью 5 и 3 литра. Наберите в один из кувшинов ровно 4 литра воды, полностью наполняя кувшины и переливая воду из одних кувшинов в другие.

36. Ограниченное разнообразие

Найдите все значения n, для которых можно заполнить таблицу размером $n \times n$ знаками «+» и «−» (по одному знаку в каждой ячейке) таким образом, чтобы для каждой ячейки ровно в одной соседней ячейке был противоположный знак. Две ячейки считаются соседними, если они находятся рядом в одной строке или в одном столбце.

37. Задача о $2n$ шашках

Для каждого $n > 1$ разместите $2n$ шашек на доске размером $n \times n$ так, чтобы в каждой строке, столбце или диагонали находилось не более двух шашек.

38. Замощение плитками тетрамино

Есть пять типов тетрамино — плиток, состоящих из четырёх квадратов размера 1×1. Они показаны на рис. 2.10.

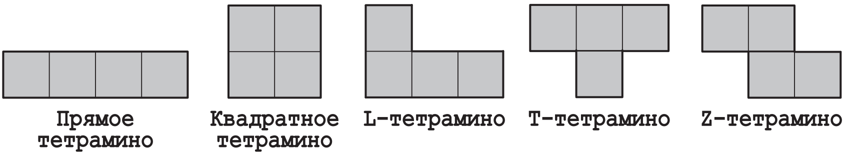


Рис. 2.10. Пять типов тетрамино

Возможно ли замостить (т. е. покрыть без наложения) шахматную доску 8×8 следующим образом:

а) 16 прямыми тетрамино;
б) 16 квадратными тетрамино;
в) 16 L-тетрамино;
г) 16 T-тетрамино;
д) 16 Z-тетрамино;
е) 15 T-тетрамино и одним квадратным?

39. Прогулки по клеточному полю

Для каждой доски (рис. 2.11) или найдите путь, который проходит через каждую клетку доски только по одному разу, или докажите, что такого пути не существует. Путь может проходить по любым соседним клеткам по горизонтали или вертикали. Возвращаться в исходную точку необязательно.

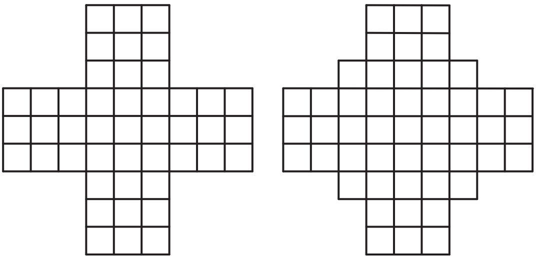


Рис. 2.11. Две доски, для которых нужно найти путь, проходящий через все клетки

40. Перестановка четырёх коней

На шахматной доске размером 3×3 стоят четыре коня. Два белых коня — в двух нижних углах, а два чёрных — в двух верхних. Найдите кратчайшую последовательность ходов, приводящую коней в положение, изображённое справа на рис. 2.12, или докажите, что это невозможно. Разумеется, два коня ни в какой момент не могут занимать одно поле на доске.

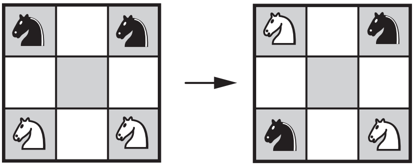


Рис. 2.12. Головоломка «*Перестановка четырёх коней*»

41. Круг света

Есть $n > 2$ светильников, расположенных по кругу. Рядом с каждым находится тумблер. Тумблер может быть в двух позициях, включая или выключая три светильника: ближайший и два соседних. Изначально все светильники выключены. Разработайте алгоритм, чтобы включить все светильники с помощью минимального числа тумблеров.

42. Ещё раз о волке, козе и капусте

У вас есть $4n$ фишек четырёх типов: n «волков», n «коз», n «кочанов капусты» и n «охотников». Расставьте фишки в ряд таким образом, чтобы все были в безопасности, т. е. рядом с волком не было бы охотника, рядом с козой — волка, рядом с капустой — козы. Кроме того, нужно, чтобы две фишки одного типа не находились рядом друг с другом. Сколько решений имеет эта головоломка?

43. Расстановка чисел

Даны n различных целых чисел и ряд из n коробок. Между коробками фиксированно расположены знаки неравенств. Придумайте алгоритм, который помещает числа в коробки так, чтобы неравенства были правильные. Например, числа 2, 5, 1 и 0 могут быть помещены в четыре коробки таким образом:

$$\boxed{0} < \boxed{5} > \boxed{1} < \boxed{2}$$

44. Легче или тяжелее?

У вас есть $n > 2$ монет, которые выглядят одинаково, и чашечные весы без гирь. Одна из монет — фальшивая, но вы не знаете, легче она или тяжелее, чем настоящие монеты. Все настоящие монеты весят одинаково. Разработайте алгоритм, позволяющий за минимальное количество взвешиваний определить, легче или тяжелее фальшивая монета.

45. Самый короткий путь коня

За какое минимальное количество ходов шахматный конь может пройти из одного угла доски размером 100×100 в угол, противоположный ему по диагонали?

46. Фишки трёх цветов

На прямоугольной доске, в которой три горизонтальных ряда и n вертикальных, стоят $3n$ фишек — n красных, n белых и n синих. Переставьте фишки таким образом, чтобы в каждом вертикальном ряду фишки были разных цветов. Разрешается переставлять фишки только в горизонтальном ряду. Придумайте алгоритм для решения задачи или докажите, что такой алгоритм не существует.

47. Планировка выставки

В музее есть выставочное пространство — 16 залов. План залов представлен на рис. 2.13. Между всеми соседними залами есть дверь. Кроме того, у каждого зала, расположенного с северной и южной сторон (верхний и нижний ряды на плане), есть ещё дверь наружу. Куратор планирующейся выставки должен решить, какие двери нужно открыть, чтобы посетитель мог войти через дверь на северной стороне, посетить каждый зал ровно один раз и выйти через дверь на южной стороне. Куратор хочет оставить открытыми минимально возможное число дверей.

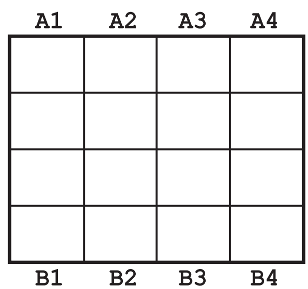


Рис. 2.13. План расположения 16 залов

а) Каково минимальное количество дверей, которые нужно оставить открытыми?

б) Какие двери нужно открыть для входа и выхода? Укажите все возможные пары дверей для входа и выхода.

48. Макнаггет-числа

Макнаггет-число — это натуральное число, которое может быть получено суммированием куриных макнаггетсов, заказанных в «Макдональдсе». Макнаггетсы продаются в коробочках по 4, 6, 9 и 20 штук.

а) Найдите все натуральные числа, которые *не* являются макнаггет-числами.

б) Разработайте алгоритм для подсчёта количества коробочек, необходимых для сервировки заказа из произвольного макнаггет-числа макнаггетсов.

49. Миссионеры и каннибалы

Три миссионера и три каннибала хотят пересечь реку. У них есть лодка, вмещающая двоих людей. Лодка сама по себе, без людей, пересечь реку не может. Грести умеют все миссионеры и все каннибалы. Если на берегу есть миссионеры, то их количество ни в какой момент времени не должно быть меньше количества каннибалов. Как всем шестерым поскорее перебраться через реку?

50. Последний шар

а) У вас в мешке 20 чёрных шаров и 16 белых. Вы вынимаете одновременно два шара. Если они одного цвета, вы добавляете в мешок чёрный шар. Если они разных цветов — добавляете белый. Повторяете это действие до тех пор, пока в мешке не останется один шар. Можно ли предсказать, какого цвета будет последний шар в мешке?

б) Ответьте на этот же вопрос, если изначально в мешке 20 чёрных шаров и 15 белых.

ГОЛОВОЛОМКИ СРЕДНЕЙ СЛОЖНОСТИ

51. Недостающее число

Джилл поспорила с Джеком, что она сможет сделать такой трюк: Джек назовёт 99 различных чисел от 1 до 100 в случайном порядке, а она сможет назвать недостающее число. Как лучше всего проделать этот трюк? Разумеется, Джилл должна решить задачу в уме, не делая записей.

52. Подсчёт треугольников

Алгоритм начинается с одного равностороннего треугольника. На каждой следующей итерации со всех сторон добавляются новые треугольники. Результаты для первых трёх итераций показаны на рис. 2.14. Сколько маленьких треугольников будет после n-й итерации?

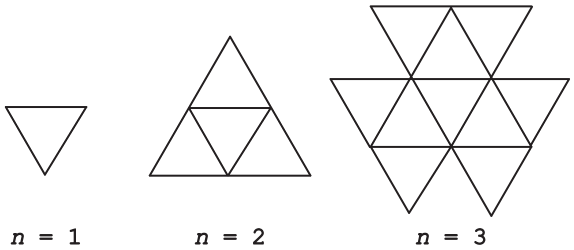


Рис. 2.14. Первые итерации алгоритма для головоломки «*Подсчёт треугольников*»

53. Определение фальшивой монеты с помощью электронных весов

У вас есть $n > 1$ монет, которые выглядят одинаково. Из них $n - 1$ монет настоящие; каждая весит g грамм. Одна из монет — фальшивая, её вес неизвестен и не равен g. Придумайте алгоритм для определения фальшивой монеты с помощью электронных весов за минимальное число взвешиваний. (Считается, что весы показывают точный вес монеты.)

54. Разрезание прямоугольника

На линованной бумаге размером $m \times n$ нарисован прямоугольник. Разрежьте его на mn квадратиков размером 1×1, делая прямые разрезы вдоль линий. Вы можете складывать несколько листов вместе и резать их одновременно. Это будет считаться за одно разрезание. Разработайте алгоритм, позволяющий выполнить задачу с минимальным числом разрезаний.

55. Головоломка «Одометр»

Одометр в машине показывает шестизначные числа от 000 000 до 999 999 включительно. Если он пройдёт через весь диапазон значений, в скольких числах будет встречаться цифра 1 по меньшей мере один раз? Сколько всего раз встретится цифра 1? (Например, число 101 111 добавляет пять единиц к общему счёту. Следующее значение одометра 101 112 добавляет ещё четыре.)

56. Строй новобранцев

Бравому солдату Швейку приказали выстроить новобранцев в ряд, чтобы офицер мог произнести перед ними речь.

Нужно построить их таким образом, чтобы минимизировать среднюю разницу в росте между новобранцами, стоящими рядом. Швейк поставил самого высокого первым, самого низкого — последним, а остальным велел встать между ними кто как захочет. Выполнил ли Швейк приказ? А как бы вы выстроили новобранцев?

57. Задача Фибоначчи о кроликах

В огороженном месте имеется пара кроликов. Эти кролики (самец и самка) только родились. Кролики в течение первого месяца своей жизни ещё не способны к размножению, но в конце второго и каждого последующего месяца дают жизнь новой паре кроликов (самец и самка). Сколько пар кроликов будет через год?

58. Сортируем раз, сортируем два...

Перетасуйте колоду из 52 карт и разложите их лицом вверх в 4 горизонтальных ряда и 13 вертикальных. Отсортируйте каждый горизонтальный ряд по старшинству, т. е. в порядке неубывания числового значения (туз — это единица, валет, королева и король — 11, 12, 13 соответственно). Для карт одинакового числового значения назначьте произвольно старшинство мастей. Например — трефы (младшие), затем бубны, черви и пики (старшие). Затем отсортируйте каждый вертикальный ряд. Если вы захотите снова отсортировать каждый горизонтальный ряд, какое максимальное количество пар карт придётся переставить?

59. Шапки двух цветов

В некоей тюрьме сидят 12 очень умных заключённых. Чтобы от них избавиться, надзиратель решает провести такое испытание. На голову каждого заключённого надели шапку чёрного или белого цвета. Есть хотя бы одна шапка каждого цвета, и заключённые об этом знают. Они могут видеть шапки на других заключённых, но свою видеть не могут. Между заключёнными нет возможности никакой коммуникации. Надзиратель собирается выстраивать заключённых в ряд каждые 5 минут с 12:05 до 12:55. Чтобы пройти испытание, заключённые в чёрных шапках (и только они) должны будут сделать шаг вперёд во время одного из построений. Если они смогут это сделать, всех заключённых освободят, если нет — казнят. Смогут ли заключённые пройти испытание?

60. Переделка треугольника из монеток в квадрат

Имеется прямоугольный равнобедренный треугольник, составленный из $n > 1$ линий, выложенных $1, 3, \ldots, 2n-1$ одинаковыми монетками соответственно (см. пример такого треугольника при $n = 3$ на рис. 2.15). Каково наименьшее число монеток, которые нужно передвинуть, чтобы сложить из всех монеток квадрат? Сколько разных квадратов можно сложить с минимальным числом передвижений?

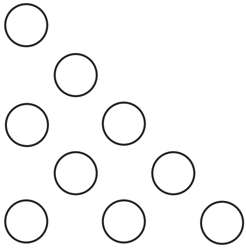


Рис. 2.15. Пример головоломки «*Переделка треугольника из монеток в квадрат*» при $n = 3$

61. Шашки на диагонали

На шашечной доске размером $n \times n$, где $n \geqslant 4$, на каждой клетке вдоль главной диагонали от верхнего левого до нижнего правого угла стоит шашка. На каждом шаге вы можете выбрать любую пару шашек и сдвинуть их на одну клетку вниз (так, чтобы шашки оставались на доске). Передвиньте все шашки на нижний ряд доски. Найдите все значения n, для которых это возможно, и придумайте алгоритм для решения задачи. Сколько шагов в алгоритме?

62. Робот собирает монетки

На доске размером $n \times m$ разложены монетки, одна монетка на каждой клетке. Робот, стоящий в верхнем левом углу доски, должен собрать как можно больше монеток и принести их в нижний правый угол. Робот может двигаться только на одну клетку вправо или на одну вниз. Когда робот приходит на клетку с монеткой, он монетку подбирает. Найдите алгоритм, позволяющий роботу собрать максимальное количество монет. Определите путь робота.

63. Плюсы и минусы

В ряд написаны n последовательных целых чисел от 1 до n. Придумайте алгоритм, который ставит перед числами знаки

«+» и «−» таким образом, чтобы в итоге полученное выражение равнялось 0. Если задача невыполнима, алгоритм выдаёт сообщение «решения нет». Ваш алгоритм должен быть более эффективным, чем перебор всех возможных вариантов расстановки знаков.

64. Восьмиугольники

На плоскости есть 2000 точек, из которых никакие три не лежат на одной прямой. Разработайте алгоритм для того, чтобы нарисовать 250 восьмиугольников с вершинами в этих точках. Восьмиугольники должны быть простыми, т. е. граница восьмиугольника не должна пересекать саму себя, и у них не должно быть общих точек.

65. Угадывание кода

Ваш друг задумал код. Кодом является n-битовая строка (некоторая последовательность нулей и единиц, например 01011 для $n = 5$). Разгадайте код, задавая другу вопросы. Каждый вопрос — это n-битовая строка, придуманная вами. Ваш друг должен сказать, сколько битов в вашей строке совпадает с соответствующими битами в коде. Например, если код 01011, а ваша строка 11001, ответ друга «три», поскольку в двух строках одинаковые биты во второй, третьей и пятой позициях. Разработайте алгоритм, который определяет любой n-битовый код не более чем за n вопросов.

66. Оставшееся число

На доске написаны первые 50 натуральных чисел: $1, 2, \ldots, 50$. Повторите 49 раз следующую операцию: выберите два числа a и b, напишите на доске абсолютную величину их разности $|a - b|$ и затем сотрите a и b. Найдите все возможные значения оставшегося числа.

67. Разливаем пополам

Есть 10 одинаковых сосудов; в одном налито a литров воды, а остальные пустые. Разрешено брать по два сосуда и распределять поровну общее количество воды в них. Цель — с помощью таких переливаний оставить минимальное количество воды в том сосуде, в котором сначала была налита вода. Как лучше это сделать?

68. Сумма цифр

Без помощи компьютера или калькулятора найдите общую сумму цифр во всех целых числах от 1 до 1 000 000 включительно.

69. Фишки на секторах круга

Круг разделён на $n > 1$ секторов, на каждый сектор кладётся фишка. На каждом ходе две фишки сдвигаются на соседний сектор (в одном направлении или в противоположных). Для каких значений n существует алгоритм, позволяющий собрать все фишки в одном секторе?

70. Прыжки в пары — 1

В ряд разложены n монеток. Сформируйте из них $\frac{n}{2}$ пар с помощью последовательности ходов. На каждом ходе монетка может перепрыгивать через две ближайшие монетки вправо или влево (либо через две одиночные монетки, либо через уже сформировавшуюся пару). После прыжка монетка «приземляется» на другую одиночную монетку. По три монетки собирать нельзя. Пустое пространство между соседними монетками игнорируется. Определите все значения n, при которых у задачи есть решение, и разработайте алгоритм, который для таких n даёт решение за минимальное число ходов.

71. Помеченные ячейки — 1

Пометьте n ячеек на бесконечном листе линованной бумаги таким образом, чтобы у каждой помеченной ячейки число помеченных соседних ячеек было положительным и чётным. Две ячейки считаются соседними, если они расположены рядом друг с другом по горизонтали или вертикали, но не по диагонали. Помеченные ячейки должны образовывать непрерывную область, т. е. между любыми двумя помеченными ячейками должен быть путь через последовательность помеченных соседних ячеек. Например, решение при $n = 4$ показано на рис. 2.16. Для каких значений n существует решение?

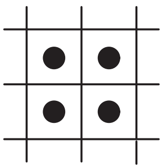


Рис. 2.16. Четыре помеченные ячейки с чётным числом помеченных соседних ячеек

72. Помеченные ячейки — 2

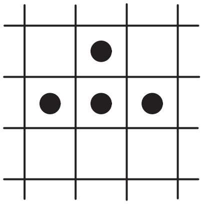


Рис. 2.17. Четыре помеченные ячейки с нечётным числом помеченных соседних ячеек

Пометьте n ячеек на бесконечном листе линованной бумаги таким образом, чтобы у каждой помеченной ячейки число помеченных соседних ячеек было нечётным. Две ячейки считаются соседними, если они расположены рядом друг с другом по горизонтали или вертикали, но не по диагонали. Помеченные ячейки должны образовывать непрерывную область, т. е. между любыми двумя помеченными ячейками должен быть путь через последовательность помеченных соседних ячеек. Например, решение при $n = 4$ показано на рис. 2.17. Для каких значений n существует решение?

73. Погоня за петухом

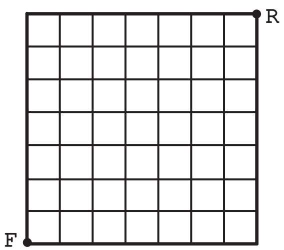


Рис. 2.18. Начальное положение в игре *«Погоня за петухом»*

Поле для игры изображено на рис. 2.18. Буквой F в левом нижнем углу обозначен фермер, буквой R обозначен петух. Фермер и петух ходят по очереди, пока фермер не поймает петуха. На каждом ходе каждый из них перемещается в соседний узел сетки вверх, вниз, направо или налево. Петух пойман, когда фермер может переместиться в узел сетки с петухом.

а) Может ли фермер поймать петуха, если фермер ходит первым? Если да, составьте алгоритм, который делает это за минимальное число ходов. Если нет, объясните почему.

б) Может ли фермер поймать петуха, если фермер ходит вторым? Если да, составьте алгоритм, который делает это за минимальное число ходов. Если нет, объясните почему.

Естественно, петух не хочет, чтобы его поймали.

74. Выбор места

Рассмотрим общий случай головоломки «*Где разместить киоск с лимонадом*», которая обсуждалась в первой части учебного раздела. Зададим местоположения n домов в городе с идеально горизонтальными улицами и вертикальными проспектами координатами (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n)$ (см. рис. 1.10, *а* в первой части учебного раздела). Придумайте алгоритм, чтобы определить местоположение такой точки (x, y), что сумма манхэттенских расстояний от всех домов до этой точки минимальна. Расстояния рассчитываются по формуле $|x_1 - x| + |y_1 - y| + \cdots + |x_n - x| + |y_n - y|$.

75. Инспекция бензоколонок

Инспектору нужно проверить работу $n > 1$ бензоколонок, расположенных на равных расстояниях друг от друга вдоль прямого шоссе. Бензоколонки пронумерованы последовательно от 1 до n. Инспектор начинает с бензоколонки 1, которую ему нужно будет посетить ещё один раз. Ему также нужно дважды посетить бензоколонку n. Там он может и закончить свой обход, хотя это не является обязательным условием. Бензоколонки с номерами от 2 до $n - 1$ он должен проверить равное количество раз. Например, он может проехать от бензоколонки 1 к бензоколонке n, поехать обратно и вернуться на бензоколонку 1, а затем поехать снова на бензоколонку n, завершив тем самым работу. (Предполагается, естественно, что инспектор посещает бензоколонки всякий раз, когда проезжает мимо них.) Вопрос: является ли этот путь кратчайшим из путей, отвечающих заданным условиям? Если да, докажите. Если нет, найдите кратчайший.

76. Быстрая ладья

Шахматная ладья ходит на любую клетку, расположенную на той же горизонтали или вертикали, на которой она стоит. За какое минимальное число ходов ладья пройдёт по всем клеткам шахматной доски размером $n \times n$? (Необязательно начинать и заканчивать на одной и той же клетке. Начальная и конечная клетки считаются пройденными по умолчанию.)

77. Поиск закономерности

Умножьте следующие числа: 1×1, 11×11, 111×111, 1111×1111.

Какая закономерность обнаруживается в произведениях? Продолжится ли эта закономерность если умножать числа с бо́льшим количеством единичек?

78. Замощение прямыми тримино

Прямое тримино представляет собой плитку размером 3×1. Очевидно, что можно замостить прямыми тримино доску размером $n \times n$, если n кратно 3. Верно ли, что для каждого $n > 3$, которое не делится на 3, можно замостить доску $n \times n$ прямыми тримино и одной плиточкой размером 1×1, которая называется мономино? Если это возможно, объясните как; если нет, объясните почему.

79. Дверцы шкафчиков

В коридоре есть n запирающихся шкафчиков, пронумерованных последовательно от 1 до n. Изначально все дверцы заперты. Вы проходите мимо шкафчиков n раз, каждый раз начиная со шкафчика № 1. Во время i-го прохода, $i = 1, 2, \ldots, n$, вы поворачиваете ключ в дверце каждого i-го шкафчика: если дверца заперта, вы отпираете её; если отперта — запираете. Таким образом, после первого прохода все дверцы отперты. Во время второго прохода вы поворачиваете ключ только в дверцах шкафчиков с чётными номерами (№ $2, 4, \ldots$), т. е. после второго прохода чётные дверцы заперты, а нечётные отперты. Проходя в третий раз, вы запираете дверцу шкафчика № 3 (которую отперли во время первого прохода), отпираете дверцу шкафчика № 6 (запертую во время второго прохода) и т. д. Какие шкафчики будут открыты и какие закрыты после последнего прохода? Сколько открытых?

80. Прогулка принца

Рассмотрим вымышленную шахматную фигуру — назовём её принцем, — которая может ходить на одну клетку вправо, или на одну вниз, или на одну по диагонали вверх и влево. Найдите все значения n, при которых принц может пройти по всем клеткам шахматной доски размером $n \times n$ в точности по одному разу во время одной прогулки.

81. Ещё раз о задаче о знаменитости

В группе из n человек есть знаменитость — тот, кто никого не знает, но которого знают все. Задача состоит в том, чтобы обнаружить знаменитость, задавая людям один вопрос «Вы её/его знаете?» Найдите алгоритм, позволяющий обнаружить знаменитость или определить, что в группе знаменитости нет. Какое максимальное количество вопросов необходимо задать, чтобы решить задачу для группы из n человек?

82. Вверх орлом

В ряд выложены n монет, в случайном порядке расположенные вверх орлом или решкой. На каждом шаге вы можете перевернуть любое количество монет, лежащих по соседству друг с другом. Придумайте алгоритм, позволяющий перевернуть все монеты вверх орлом за минимальное количество шагов. Сколько шагов потребуется при самом плохом сценарии?

83. Ханойская башня с ограничением

Есть n колец различного размера и три стержня. Вначале все кольца расположены на первом стержне в порядке возрастания размера, самое большое внизу, а самое маленькое наверху. Задача — переместить все кольца на третий стержень. За один раз можно переместить только одно кольцо; класть больший диск на меньший нельзя. Помимо этого, на каждом шаге можно либо поместить кольцо на средний стержень, либо убрать с него (рис. 2.19). Придумайте алгоритм, решающий задачу за минимальное количество шагов.

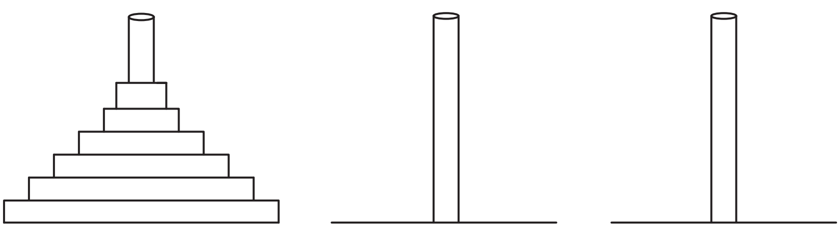


Рис. 2.19. Головоломка «*Ханойская башня с ограничением*». Переместить все кольца с левого стержня на правый посредством среднего стержня

84. Сортируем блины

Друг на друге лежат n блинов разного размера. Вы можете просунуть кухонную лопаточку между блинами и перевернуть разом всю горку блинов, оказавшихся на лопаточке. Задача: отсортировать блины по размеру так, чтобы самый большой был внизу. Рисунок 2.20 показывает частный случай головоломки при $n = 7$. Придумайте алгоритм для решения этой задачи и определите необходимое число переворотов при худшем сценарии.

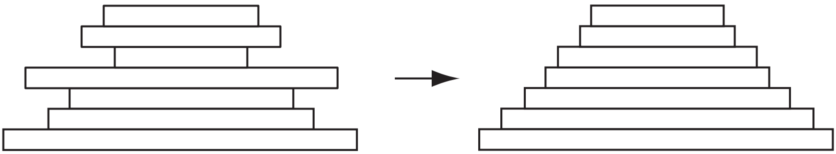


Рис. 2.20. Головоломка «*Сортируем блины*» при $n = 7$

85. Распространение сплетен — 1

Есть n человек, каждый из которых знает какую-нибудь сплетню, которую не знает никто другой. Они хотят рассказать эти сплетни друг другу по электронной почте. Какое минимальное количество сообщений им нужно послать, чтобы каждый знал все сплетни? Предположим, что каждый посылающий сообщение рассказывает обо всех известных ему сплетнях и отправляет его только по одному адресу.

86. Распространение сплетен — 2

Есть n человек, каждый из которых знает какую-нибудь сплетню, которую не знает никто другой. Они хотят рассказать эти слухи друг другу по телефону (т. е. когда разговаривают только два человека). Разработайте для решения этой задачи эффективный (с точки зрения общего числа разговоров) алгоритм. Предположим, что в каждом разговоре собеседники обмениваются всеми известными им на момент разговора сплетнями.

87. Перевёрнутые стаканы

На столе стоят n стаканов, все перевёрнутые. На каждом шаге вы можете перевернуть в точности $n - 1$ стаканов. Определите все значения n, для которых все стаканы можно поставить правильно. Найдите алгоритм, который делает это за минимальное число шагов.

88. Жабы и лягушки

На одномерной доске размером $2n + 1$ ячеек в первых ячейках стоят n фишек, представляющих n жаб, а в последних n ячейках — n фишек, представляющих лягушек. Жабы и лягушки двигаются по очереди либо переходя в соседнюю пустую ячейку, либо перепрыгивая в неё через одного представителя другого вида. (Жабы не могут прыгать через жаб, лягушки — через лягушек.) Жабы могут передвигаться только направо, лягушки — только налево. Задача — поменять местами жаб и лягушек. Например, для $n = 3$ задачу можно представить такой схемой, где T — это жаба, а F — лягушка:

T	T	T		F	F	F

$\Longrightarrow$

F	F	F		T	T	T

Разработайте алгоритм для решения задачи.

89. Перестановка фишек

В этот пасьянс играют на двумерной доске из $2n+1$ строк и $2n+1$ столбцов. На всех $(2n+1)^2$ клетках доски, кроме центральной, стоят фишки двух цветов — например, белого (W) и чёрного (B) в следующем порядке. В первых n рядах на первых $n+1$ позициях стоят фишки W, за ними n фишек B. В ряду $n+1$ на первых n позициях стоят фишки W, за ними — одна пустая позиция и n фишек B. В последних n строках на первых n позициях стоят n фишек W, за ними стоят $n+1$ фишек B. Фишки W могут двигаться по горизонтали направо или по вертикали вниз; фишки B — по горизонтали налево или по вертикали наверх. Ход — это либо перемещение на пустую соседнюю позицию, либо прыжок через одну фишку противоположного цвета на пустую позицию непосредственно за ней. Фишки не могут перепрыгивать через фишки своего цвета. Цель — переместить все фишки на позиции, на которых изначально стояли фишки противоположного цвета (см. рис. 2.21 для $n = 3$).

W	W	W	W	B	B	B
W	W	W	W	B	B	B
W	W	W	W	B	B	B
W	W	W		B	B	B
W	W	W	B	B	B	B
W	W	W	B	B	B	B
W	W	W	B	B	B	B

$\longrightarrow$

B	B	B	B	W	W	W
B	B	B	B	W	W	W
B	B	B	B	W	W	W
B	B	B		W	W	W
B	B	B	W	W	W	W
B	B	B	W	W	W	W
B	B	B	W	W	W	W

Рис. 2.21. Перестановка фишек для $n = 3$

90. Пересаживания

На n стульях, расположенных в ряд, сидят n детей. Можно ли придумать такой алгоритм пересаживания детей, чтобы при его применении получились все возможные варианты их размещения, если пересаживаться можно только детям, которые сидят рядом друг с другом?

91. Горизонтальные и вертикальные домино

Найдите все значения n, для которых доску размером $n \times n$ можно замостить одинаковым числом вертикально и горизонтально лежащих домино.

92. Замощение трапециями

Равносторонний треугольник разделён на маленькие равносторонние треугольники с помощью параллельных прямых, разбивающих каждую его сторону на $n > 1$ равных частей. Самый верхний треугольник отрезают; пример получающейся фигуры при $n = 6$ изображён на рис. 2.22. Эту фигуру требуется покрыть трапециями, составленными из трёх равносторонних треугольников того же размера, что и треугольники, из которых состоит фигура (трапеции не обязаны быть одинаково повёрнуты, но должны покрывать всю фигуру без наложений). Найдите все n, для которых это возможно, и разработайте алгоритм замощения для таких n.

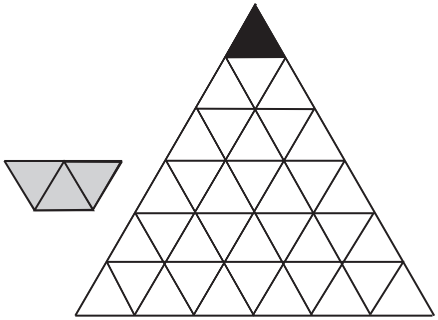


Рис. 2.22. Фигура, которую нужно замостить трапециями (трапеция нарисована серым), для $n = 6$

93. Стрельба по линкору

За какое минимальное число выстрелов игрок обязательно попадёт в линкор (прямоугольник 4×1) на поле 10×10? Линкор может быть расположен горизонтально или вертикально в любом месте доски. Будем считать, что других кораблей нет. («Выстрел» — это наугад названная клетка поля.)

94. Поиск в отсортированном массиве

Сто разных номеров написаны на 100 картах — по числу на карту. Карты расположены в 10 рядов и 10 столбцов, по возрастанию чисел в каждом ряду (слева направо) и в каждом столбце (сверху вниз). Все карты переворачивают так, чтобы числа на них не были видны. Существует ли алгоритм, с помощью которого можно узнать, написан ли заданный номер на какой-либо из карт, после переворачивания не более 20 карт?

95. Максимальный и минимальный вес

Даны $n > 1$ предметов и чашечные весы без гирь. Определить самый лёгкий и самый тяжёлый предметы за $\left\lceil \frac{3n}{2} \right\rceil - 2$ взвешивания.

96. Замощение лестницы

Найдите все значения $n > 1$, для которых фигура S_n — «лестница» (см. рис. 2.23 для $n = 8$) — может быть замощена угловыми тримино.

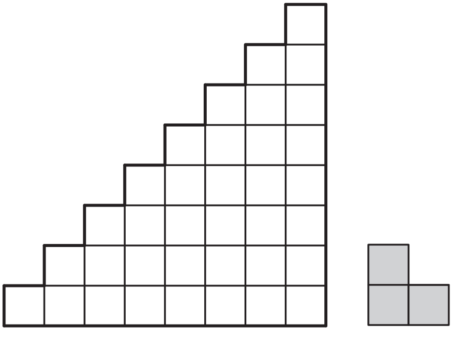


Рис. 2.23. Фигура S_8 — «лестница», которую требуется покрыть угловыми тримино (тримино закрашено серым)

97. Обмен в колоде карт

Рассмотрим следующую карточную игру для одного. В неё играют 13 картами одной масти; каждой карте присвоено значение: туз — 1, двойка — 2, и т. д., валет, дама, король — соответственно 11, 12, 13. Перед началом игры карты перетасовывают. После этого повторяется такая процедура. Верхнюю карту в колоде поворачивают лицом вверх. Если это туз — игра заканчивается. В противном случае верхние n карт, где n — значение верхней карты, убирают из колоды и возвращают в неё в обратном порядке. Пример хода в такой игре:

$$5\ 7\ 10\ K\ 8\ A\ 3\ Q\ J\ 4\ 9\ 2\ 6 \implies$$
$$\implies 8\ K\ 10\ 7\ 5\ A\ 3\ Q\ J\ 4\ 9\ 2\ 6.$$

Заканчивается ли такая игра после конечного числа ходов при любом начальном состоянии колоды?

98. Ромбопалиндром

Сколькими способами можно прочитать палиндром

WAS IT A CAT I SAW

в ромбе, изображённом на рис. 2.24? Можно начинать с любого W и на каждом шаге двигаться в любом направлении — вверх, вниз, влево или вправо — к соседней букве. Каждую букву можно использовать несколько раз в одной последовательности ходов.

```
            W
          W A W
        W A S A W
      W A S I S A W
    W A S I T I S A W
  W A S I T A T I S A W
W A S I T A C A T I S A W
  W A S I T A T I S A W
    W A S I T I S A W
      W A S I S A W
        W A S A W
          W A W
            W
```

Рис. 2.24. Расположение букв в головоломке «*Ромбопалиндром*»

99. Обратная сортировка

В ряд расположены n карт, на которых написаны n различных целых чисел-номеров (по одному на карту). Карты лежат в порядке убывания номеров. Разрешается менять местами любые две карты, между которыми лежит ровно одна карта. Для каких n с помощью такой операции можно пересортировать карты так, чтобы они лежали в порядке возрастания номеров? Найдите алгоритм, требующий минимального числа обменов.

100. Куда доскачет конь

Сколько существует различных клеток, на которых может оказаться шахматный конь после n ходов, сделанных на бесконечной доске? (Конь ходит буквой Г: две клетки вверх, вниз, влево или вправо и затем на одну клетку в перпендикулярном направлении.)

101. Перекраска комнат

Жил-был король, который любил шахматы. У него был дворец, подобный шахматной доске 8×8. Во дворце было 64 комнаты, и в каждой комнате в каждой из четырёх стен была дверь. Сначала пол во всех комнатах был белым. Потом король приказал перекрасить полы, чтобы вместе они напоминали настоящую шахматную доску (рис. 2.25). Маляр был обязан, проходя по дворцу, перекрашивать пол в каждой комнате, в которой он оказывался, из белого в чёрный цвет и наоборот. Маляру разрешалось выходить из дворца и заходить в него в другом месте. Была ли возможность исполнить приказ так, чтобы перекрашивать комнаты не более 60 раз?

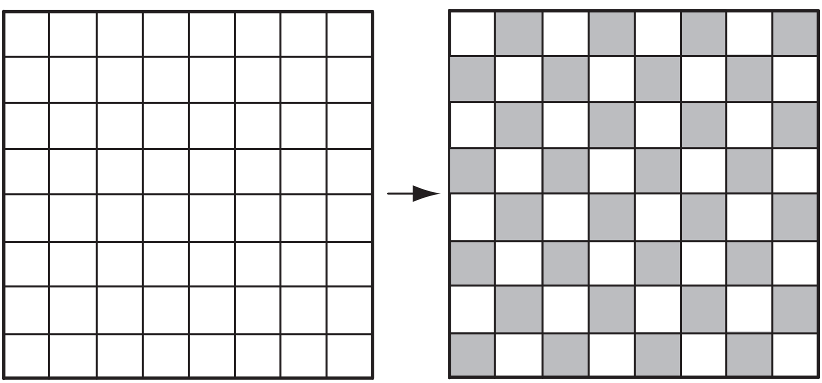


Рис. 2.25. Задача о перекраске комнат

102. Обезьянка и кокосовые орехи

Пять моряков и обезьяна потерпели кораблекрушение и были выброшены на необитаемый остров. В первый день они набрали кокосовых орехов, чтобы съесть их утром. Ночью один моряк проснулся, дал один кокос обезьянке, поделил оставшиеся кокосы на пять равных частей, спрятал свою часть, сложил остальные кокосы обратно в кучу и пошёл спать. Позже ночью остальные четверо моряков сделали то же самое друг за другом: каждый отдал один орех обезьянке и взял себе одну пятую часть из оставшихся. Утром они разделили между собой оставшиеся орехи, дав обезьянке орех ещё раз. Какое минимальное число орехов могло быть первоначально?

103. Прыжки на другую сторону

На 15 клетках доски размером 5×6, обозначенных чёрным цветом на рис. 2.26, стоят фишки. Задача — переместить все фишки, которые стоят выше черты, на клетки ниже черты. На каждом ходе фишка может перепрыгивать через соседнюю фишку на незанятую клетку, расположенную сразу же за ней. Прыгать можно по горизонтали, вертикали и любой диагонали. Выполнима ли задача?

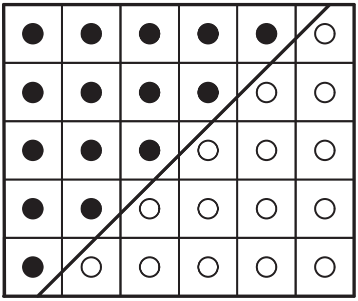


Рис. 2.26. Доска для головоломки «*Прыжки на другую сторону*»

104. Разделение кучи фишек

а) В куче лежат n фишек. Разделите фишки на две кучки и подсчитайте произведение количества фишек в этих двух кучках. Продолжайте делить каждую кучку на две и подсчитывать произведения до тех пор, пока не останется n кучек по одной фишке. После этого просуммируйте все полученные произведения. Как нужно делить кучки, чтобы сумма произведений была максимальна? Чему равна эта максимальная сумма?

б) Как изменится решение головоломки, если мы будем подсчитывать сумму количества фишек в двух кучках после каждой делёжки и если нам нужно максимизировать сумму этих сумм?

105. Головоломка «MU»

Рассмотрим цепочки, составленные из трёх символов M, I и U, которые получаются из начальной цепочки MI применением конечного числа следующих допустимых преобразований.

1. Можно добавить U в конец каждой цепочки, заканчивающейся на I; например, изменить MI на MIU.

2. Можно удвоить любую последовательность символов, следующих после M (т. е. изменить Mx на Mxx); например, изменить MIU на MIUIU.

3. Можно заменить последовательность III на U; например, изменить MUIIIU на MUUU.

4. Можно убрать UU. Например, изменить MUUU на MU.

Возможно ли, действуя по этим правилам, получить цепочку MU?

106. Лампочка и переключатели

Лампочка соединена с n переключателями таким образом, что она загорается только в том случае, когда все переключатели замкнуты. Каждый переключатель управляется кнопкой; нажатие кнопки меняет положение переключателя. Узнать, в каком положении находится переключатель, невозможно. Разработайте алгоритм, позволяющий включать лампочку за минимальное число нажатий на кнопки в наихудшем случае.

107. Лиса и заяц

Поиграем в игру-догонялку под названием «Лиса и заяц». В неё играют на одномерной доске с 30 клетками, пронумерованными слева направо от 1 до 30. Фишка-лиса начинает на клетке 1, фишка-заяц — на некоторой клетке $s > 1$. Лиса и заяц ходят по очереди, лиса начинает. На каждом ходе лиса может передвинуться на соседнюю клетку влево или вправо. Заяц прыгает влево или вправо через две клетки на третью. Заяц не может прыгнуть на клетку, где стоит лиса. Если ему некуда больше прыгать, он проиграл. Естественно, никто из них не может перемещаться за пределы доски. Цель лисы — поймать зайца, цель зайца — убежать. Найдите все значения s, при которых лиса может выиграть.

108. Самый длинный путь

Если кому-либо нужно повесить объявление на каждом из столбов, расположенных на одинаковом расстоянии друг от друга вдоль прямой дороги, то лучше всего начать с первого столба и ехать по дороге, вешая объявления на каждом столбе до последнего. Каков наихудший вариант выполнения задачи, т. е. вариант, при котором надо проехать самый длинный путь? Начинать у первого столба и заканчивать у последнего не требуется, но все повороты нужно делать у столбов.

109. Домино «дубль-n»

Домино — это прямоугольные плитки с точками на обеих половинках. Ими играют в различные настольные игры, выкладывая их определённым образом. В стандартном наборе домино «дубль-шесть» 28 плиточек-костяшек, по одной для каждой неупорядоченной пары значений от $(0, 0)$ до $(6, 6)$. В общем случае набор домино «дубль-n» состоял бы из костяшек со значениями от $(0, 0)$ до (n, n).

а) Найдите количество костяшек в наборе домино «дубль-n».

б) Найдите суммарное количество точек на всех костяшках набора домино «дубль-n».

в) Разработайте алгоритм построения кольца из всех костяшек набора домино «дубль-n» или докажите, что такого алгоритма не существует. (Разумеется, каждая пара соседних костяшек должна иметь одинаковое количество точек на прилегающих половинках.)

110. Хамелеоны

Исследователь привёз на остров три вида хамелеонов: 10 коричневых, 14 серых и 15 чёрных. Когда встречаются два хамелеона разного цвета, оба они меняют свои цвета на третий. Возможно ли, чтобы все хамелеоны стали одного цвета?

СЛОЖНЫЕ ГОЛОВОЛОМКИ

111. Переворачивание треугольника из монеток

Рассмотрим равносторонний треугольник, составленный из лежащих вплотную друг к другу одинаковых монеток (пример такого треугольника приведён на рис. 2.27. (Считаем, что центры монеток образуют равномерную треугольную решётку. Придумайте алгоритм переворачивания треугольника сверху вниз за минимальное число ходов, если за ход можно передвинуть одну монетку, не отрывая её от стола. Приведите компактную формулу для минимального числа ходов.

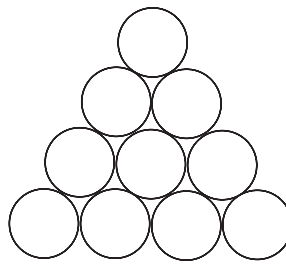


Рис. 2.27. Пример равностороннего треугольника из монеток, который нужно перевернуть сверху вниз

112. Снова о покрытии домино

Найдите все значения n, для которых шахматную доску $n \times n$ без двух полей разного цвета можно покрыть плитками домино 2×1.

113. Исчезающие монеты

Пусть на столе лежат расположенные в линию n монеток; некоторые лежат орлами (H) вверх, некоторые — решками (T), в произвольном порядке. Цель головоломки — убрать все монетки с помощью последовательности ходов. На каждом ходу можно убрать любую монету, лежащую орлом вверх; после этого соседнюю с ней монету или монеты (если они есть) нужно перевернуть. Монеты называются «соседними», если они лежат непосредственно рядом друг с другом в исходной линии; если между монетами после нескольких ходов появился свободный промежуток, такие монеты «соседними» не считаются. Например, ниже приведена последовательность ходов, которая решает задачу для конкретного начального расположения монет. (Те лежащие орлами вверх монеты, которые убираются на данном ходу, выделены жирным шрифтом.)

Т	Н	Н	Т	**Н**	Н	Н
Т	**Н**	Н	Н	_	Т	Н
Н	_	Т	Н	_	Т	Н
_	_	Т	**Н**	_	Т	Н
_	_	**Н**	_	_	Т	Н
_	_	_	_	_	Т	**Н**
_	_	_	_	_	**Н**	_
_	_	_	_	_	_	_

Определите свойство начального расположения монет, которое является необходимым и достаточным для того, чтобы головоломка имела решение. Для последовательностей монеток, которые можно убрать, не нарушая правил, придумайте соответствующий алгоритм.

114. Обход точек

Дана $(n \times n)$-сетка из точек (например, точки пересечения n последовательных горизонтальных и n последовательных вертикальных линий на обычной бумаге в клетку); $n > 2$. Проведите $2n-2$ отрезков, не отрывая карандаша от бумаги, так, чтобы пройти через все точки. Через точку можно проходить более одного раза, но по линиям — нельзя. («Жадное» решение для $n = 4$ приведено на рис. 2.28 — в нём 7 линий, а не 6, необходимые для решения головоломки).

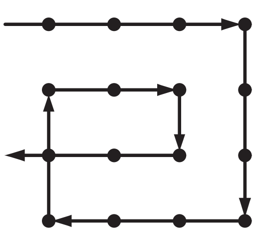


Рис. 2.28. Обход 16 точек с помощью 7 отрезков

115. Задача Баше о гирях

Найдите оптимальный набор из n гирь $\{w_1, w_2, \ldots, w_n\}$, с помощью которых можно было бы определить любой целочисленный вес груза в наибольшем возможном интервале от 1 до W. Рассмотрите два случая:

а) гири можно класть только на свободную от груза чашу весов;

б) гири можно класть на обе чаши весов.

116. Подсчёт «пустых номеров»

Если турнир на выбывание начинается с числом игроков n, не являющимся степенью двойки, то некоторым игрокам приходится «дать пустой номер», т. е. пропустить их сразу в следующий раунд, потому что им не нашлось противника. Определите общее число пустых номеров в таком соревновании, если пустые номера даются по одному из следующих правил.

а) Пустые номера даются наименьшему числу игроков в первом раунде так, чтобы в следующем раунде осталось число игроков, являющееся степенью двойки.

б) Пустые номера даются наименьшему числу игроков в каждом раунде так, чтобы в нём осталось чётное число игроков.

117. Одномерный солитёр

Рассмотрим одномерную версию игры под названием «колышковый солитёр», в который играют на поле из n клеток, где n чётно и больше двух. Изначально все клетки, кроме одной, заняты фишками («колышками»), один колышек на клетку. На каждом ходу один из колышков перепрыгивает через непосредственно соседний с ним слева или справа и ставится на свободную клетку. После прыжка колышек, через который перепрыгнули, убирается с доски. Цель — убрать все колышки, кроме одного, с помощью последовательности таких ходов. Найдите все возможные позиции изначально пустой клетки, для которых задача имеет решение, и для каждой — соответствующую позицию единственного остающегося колышка.

118. Шесть коней

На шахматной доске 3×4 расположены шесть коней: три белых в нижнем ряду и три чёрных в верхнем. Поменяйте коней местами, чтобы получить позицию, изображённую на рис. 2.29, за минимальное число ходов (так, чтобы ни на одном ходу в одной клетке не было больше одного коня).

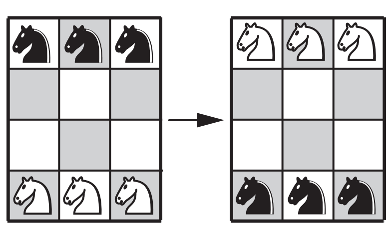


Рис. 2.29. Задача о шести конях

119. Замощение цветными тримино

Придумайте алгоритм решения следующей задачи: замостить данную доску $2n \times 2n$ $(n > 1)$ без одной клетки угловыми тримино трёх цветов так, чтобы никакие тримино одного цвета не соприкасались краями. Напомним, что угловое тримино — это L-подобная фигура из трёх квадратов (см. рис. 1.4).

120. Машина, распределяющая пенни

6				
4	1			
2	2			
0	3			
0	1	1		

Рис. 2.30. Пример распределения пенни

«Машина» состоит из ряда коробок. Вначале n пенни помещают в самую левую коробку. Потом машина перераспределяет пенни между коробками следующим образом. На каждом шаге она вынимает два пенни из коробки и кладёт одно пенни в ближайшую коробку, стоящую справа. Шаги прекращаются, когда в каждой коробке лежит не больше одной монеты. Например, рис. 2.30 показывает, как машина распределяет шесть пенни, каждый раз вынимая пару пенни из самой левой из коробок, если в ней есть хотя бы две монеты.

а) Зависит ли конечное распределение пенни от порядка, в котором машина вынимает пары монет?

б) Какое минимальное число коробок необходимо, чтобы распределить n пенни?

в) Сколько шагов сделает машина, прежде чем остановиться?

121. Проверка супер-яйца

Некая фирма изобрела суперпрочное яйцо. Она хочет определить (для рекламных целей) максимальный этаж в 100-этажном здании, упав с которого, яйцо не разобьётся. Фирма выделила человеку, ответственному за проверку, два одинаковых яйца. Разумеется, одно яйцо можно бросать много раз, пока оно не разбилось. Какое минимальное число бросков необходимо, чтобы наверняка определить самый высокий безопасный этаж?

122. Мир в парламенте

У каждого члена парламента есть максимум три врага. (Считаем, что враждебность всегда взаимна.) Правда ли, что всегда можно разделить парламент на две палаты так, чтобы у каждого парламентария было не больше одного врага в его/её палате?

123. Задача о флаге Нидерландов

В ряд расположены n шашек трёх цветов: красные, белые и синие. Придумайте алгоритм перестановки шашек в положение, при котором сначала в ряду идут все красные шашки, потом — все белые, а в конце — все синие. Единственные допустимые операции — определение цвета шашки и перестановка двух шашек.

124. Разделение цепочки

Пусть дана цепочка из $n > 1$ скрепок. Какое минимальное число одиночных скрепок необходимо убрать из цепи, чтобы из получившихся кусочков можно было составить цепь любой длины от 1 до n включительно?

125. Отсортировать 5 за 7

Есть пять предметов разного веса и чашечные весы без гирь. Отсортируйте предметы в порядке возрастания веса не более чем за семь взвешиваний.

126. Деление пирога по-честному

Друзья ($n > 1$ человек) хотят поделить между собой пирог так, чтобы каждый был доволен куском, который он получит. Разработайте алгоритм для этого.

127. Задача о ходе коня

Может ли конь пройти через все поля шахматной доски 8×8 ровно по одному разу так, чтобы путь закончился на поле, находящемся на расстоянии одного хода коня от начального? (Такой маршрут называют замкнутым. Обратите внимание, что поле считается пройденным только в том случае, когда конь встал на него, а не просто прошёл мимо в процессе хода.)

128. Тумблеры системы охраны

Дан ряд из n тумблеров, предназначенных для защиты входа на военный объект. Тумблеры работают следующим образом:

(i) Самый правый тумблер можно повернуть во включённое или выключенное положение когда угодно.

(ii) Любой другой тумблер можно включить или выключить только в том случае, если тумблер, находящийся непосредственно справа от него, включён, а остальные тумблеры справа, если они есть, выключены.

(iii) За один раз можно изменить положение только одного тумблера.

Изначально все тумблеры включены. Разработайте алгоритм для выключения всех тумблеров за минимальное число ходов. (Ходом считается одно переключение.) Также найдите это минимальное число ходов.

129. Головоломка Реве

Даны восемь колец разных размеров и четыре стержня. Изначально все кольца упорядочены по размеру на первом стержне — самое большое в самом низу, самое маленькое наверху. Цель — переместить все кольца на другой стержень с помощью последовательности ходов. За ход можно переместить только одно кольцо, причём нельзя класть большее кольцо сверху меньшего. Придумайте алгоритм, решающий задачу за 33 хода.

130. Отравленное вино

Злого короля предупредили, что вино в одной из 1000 его бочек было отравлено. Яд столь сильный, что малейшая его капля, даже сколь угодно разбавленная, убивает человека ровно через 30 дней. Король готов пожертвовать 10 рабами, чтобы определить бочку с отравленным вином.

а) Возможно ли успеть это сделать до празднества, которое состоится через 5 недель?

б) С каким минимальным числом рабов король смог бы достичь цели?

131. Задача о шашках Тэта

Дан ряд из n белых (W) и n чёрных (B) шашек без пробелов между соседними шашками. Шашки расположены в порядке чередования цветов BWBW…BW. Цель — переместить шашки так, чтобы сначала шли все белые, а затем — все чёрные, по-прежнему без пробелов между шашками: WW…WBB…B. Шашки разрешено двигать парами; на каждом ходу можно передвинуть пару соседних шашек на свободное место, не меняя их порядка. Придумайте алгоритм для решения этой задачи за n ходов для любого $n \geqslant 3$.

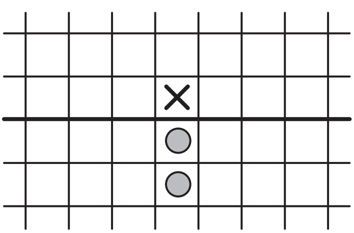
132. Солдаты Конвея

В эту версию колышкового солитёра играют на бесконечной двумерной доске с горизонтальной прямой, разделяющей доску пополам. В начальной позиции некоторое число фишек-колышков («солдаты солитёровой армии») стоят ниже этой линии. Цель — продвинуть один из колышков («разведчика» армии) как можно дальше за линию с помощью последовательности горизонтальных или вертикальных прыжков. На каждом ходу один из колышков перепрыгивает через непосредственно соседний с ним в вертикальном или горизонтальном направлении и ставится на свободную клетку; после прыжка колышек, через который перепрыгнули, убирается с доски. Например, чтобы продвинуть колышек на клетку в первом ряду за линией, хватит двух колышков (рис. 2.31, *а*); а чтобы продвинуть колышек на клетку во втором ряду, необходимо и достаточно иметь четыре колышка (рис. 2.31, *б*).

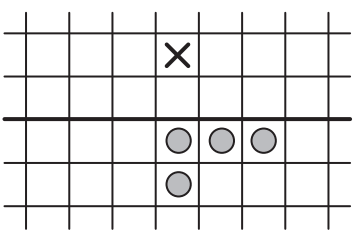
Найдите начальное расположение колышков для того, чтобы:

а) с помощью 8 колышков продвинуть 1 колышек на клетку в третьем ряду за линией;

б) с помощью 20 колышков продвинуть 1 колышек на клетку в четвёртом ряду за линией.



(*а*)



(*б*)

Рис. 2.31. Решение к задаче «*Солдаты Конвея*». (*а*) Продвижение фишки в первый ряд за «вражескую» линию. (*б*) Продвижение фишки во второй ряд за «вражескую» линию. X обозначает клетку-цель

133. Игра «Жизнь»

Игра происходит на бесконечной двумерной доске. Каждая клетка находится в одном из двух состояний: живом или мёртвом. После того как выбрана некоторая начальная конфигурация живых клеток — например, их помечают чёрной точкой — получают последовательность новых конфигураций («поколений») по следующим правилам, которые одновременно применяют к каждой клетке одного поколения. Каждая клетка взаимодействует с восемью соседями — клетками, прилегающими к ней по горизонтали, вертикали и диагонали. На каждом временно́м шаге происходят следующие перемены:

(i) *Смерть из-за недостаточной населённости.* Каждая живая клетка, у которой меньше двух живых соседей, умирает.

(ii) *Смерть из-за перенаселённости.* Каждая живая клетка, у которой больше трёх живых соседей, умирает.

(iii) *Выживание.* Каждая живая клетка, у которой два или три живых соседа, переходит в следующее поколение.

(iv) *Рождение.* Каждая мёртвая клетка, у которой ровно три живых соседа, становится живой клеткой.

а) Найдите самую маленькую начальную конфигурацию живых клеток, которая не будет меняться от поколения к поколению. (Такие конфигурации называются «натюрмортами».)

б) Найдите самую маленькую начальную конфигурацию живых клеток, которая будет переключаться между двумя состояниями. (Такие конфигурации называются «осцилляторами».)

в) Найдите самую маленькую начальную конфигурацию живых клеток, которая будет перемещать сама себя по доске. (Такие конфигурации называются «космическими кораблями».)

134. Раскраска точек

Придумайте алгоритм для решения следующей задачи. Дано n произвольных точек на сетке. Раскрасьте их в два цвета, например, чёрный и белый, так, чтобы на всех горизонтальных и вертикальных прямых числа белых и чёрных точек либо совпадали, либо отличались на единицу.

135. Разные пары

Воспитатель должен ежедневно распределять $2n$ детей на n пар для прогулок. Разработайте алгоритм для этого так, чтобы за $2n - 1$ дней пары не повторялись.

136. Поимка шпиона

В компьютерной игре шпион находится на одномерной прямой. В момент времени 0 шпион находится в точке a. За каждый временной интервал шпион перемещается на расстояние b вправо, если $b \geqslant 0$, и на $|b|$ влево, если $b < 0$. И a, и b — фиксированные целые числа, но они вам неизвестны. Ваша цель — выяснить, где находится шпион; в каждый момент времени (начиная с момента 0) вам разрешается спрашивать, находится ли шпион в некоторой выбранной вами точке. Например, вы всегда можете спросить, находится ли сейчас шпион в точке 19, и вы получите на этот вопрос правдивый ответ «да» или «нет». Если ответ «да», вы достигли своей цели; если «нет» — вы можете в следующий момент спросить, находится ли шпион в какой-то точке — той же или любой другой, по вашему выбору. Придумайте алгоритм, позволяющий найти шпиона за конечное число вопросов.

137. Прыжки в пары — 2

В ряд расположены n монет. Цель — сформировать из них $\frac{n}{2}$ пар с помощью последовательности ходов. На первом ходу некоторая монета должна перепрыгнуть через одну из непосредственно соседствующих с ней монет, на втором некоторая монета должна перепрыгнуть через две соседние, на третьем некоторая монета должна перепрыгнуть через три соседние, и т. д., пока после $\frac{n}{2}$ ходов не создадутся $\frac{n}{2}$ пар. (На каждом ходу монета может прыгнуть налево или направо, но приземлиться должна ровно на одну монету. Перепрыгивание через сформированную пару монет считается перепрыгиванием через две монеты. Свободное пространство между соседними монетами игнорируется.) Определите все значения n, при которых задача имеет решение, и придумайте алгоритм решения для таких n за минимальное число ходов.

138. Делёж конфет

В детском саду n детей сидят в кругу, в центе которого находится их воспитательница. Изначально у каждого ребёнка есть чётное число конфет. Когда воспитательница дует в свисток, каждый ребёнок одновременно с остальными отдаёт половину своих конфет соседу слева. После этого те дети, у которых оказывается нечётное число конфет, получают ещё по одной конфете от воспитательницы. Затем воспитательница снова дует в свисток, и так до тех пор, пока у всех детей не окажется одинаковое число конфет, после чего игра заканчивается. Может ли эта игра длиться бесконечно, или рано или поздно она закончится?

139. Круглый Стол короля Артура

Король Артур хочет усадить $n > 2$ рыцарей за Круглый стол так, чтобы ни один из рыцарей не сидел рядом со своим врагом. Покажите, как это сделать, если у каждого рыцаря не меньше $\frac{n}{2}$ друзей. Будем считать, что дружба и вражда всегда взаимны.

140. Снова задача об n ферзях

Рассмотрим задачу: расположить n ферзей на шахматной доске $n \times n$ так, чтобы никакие два ферзя не находились в одной строке, столбце или на одной диагонали. Придумайте линейный по сложности алгоритм для решения этой задачи для любого $n > 3$.

141. Задача Иосифа Флавия

По кругу стоят n человек, пронумерованные от 1 до n. Начиная с человека под номером 1, каждого второго человека убивают, пока не останется только один. В каком месте круга должен встать человек, чтобы остаться последним?

142. Двенадцать монет

Даны 12 одинаковых с виду монет; известно, что либо все они настоящие, либо среди них ровно одна фальшивая. Неизвестно, легче или тяжелее фальшивая монета настоящей. У вас есть чашечные весы без гирь. Задача — определить, все ли монеты настоящие, и если нет — определить фальшивую монету и узнать, тяжелее она настоящей или легче. Разработайте алгоритм для решения этой задачи за минимальное число взвешиваний.

143. Заражённая шахматная доска

Вирус распространяется по квадратам шахматной доски $n \times n$, заражая каждый квадрат, у которого есть два заражённых соседа (по горизонтали или вертикали, но не по диагонали). Какое минимальное число квадратов должно быть заражено изначально, чтобы вирус распространился на всю доску?

144. Разрушение квадратов

Рассмотрим доску $n \times n$, составленную из $2n(n+1)$ спичек, образующих границы квадратов 1×1 (см. пример на рис. 2.32). Придумайте алгоритм удаления спичек, позволяющий разрушить периметры всех квадратов любого размера за минимальное число удалений.

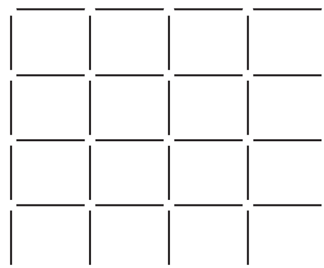


Рис. 2.32. Доска 4×4 для головоломки «*Разрушение квадратов*»

145. Пятнашки

В этой известной головоломке участвуют пятнадцать пронумерованных от 1 до 15 квадратных костяшек, заключённых в коробку 4×4 (один квадрат из шестнадцати в коробке остаётся пустым). Цель — из некоторого начального положения упорядочить костяшки по возрастанию номеров, передвигая их по одному по коробке. Можно ли решить задачу для начального положения, изображённого на рис. 2.33?

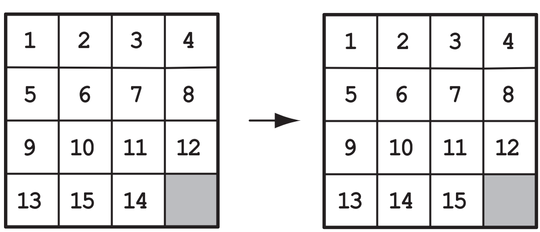


Рис. 2.33. Начальное и конечное положения игры в пятнашки

146. Стрельба по движущейся мишени

В компьютерной игре фигурируют стрелок и движущаяся мишень. Стрелок может поразить любой из $n > 1$ щитков, за которыми прячется мишень. Щитки расположены на прямой. Стрелок никогда не видит мишень; он знает лишь, что мишень всегда перемещается за соседний щиток в промежутке между двумя его последовательными выстрелами. Разработайте алгоритм, гарантирующий попадание по мишени, или докажите, что его не существует.

147. Шапки с номерами

На общеуниверситетской новогодней вечеринке было $n > 1$ математиков. Они сговорились заключить с ректором университета, который тоже посетил вечеринку, следующее пари. Ректор пишет любое число от 0 до $n-1$ включительно на каждой из шапок, которые носят математики по случаю вечеринки. Все числа могут быть различны, но это не обязательно. После того как каждый математик видит числа, написанные на всех шапках, кроме его собственной, не переговариваясь с другими математиками или кем бы то ни было ещё, он пишет на бумажке число со своей шапки и отдаёт бумажку ректору. Разумеется, ни один из них не видит, что пишут остальные. Если хотя бы одно число оказывается правильным, они выигрывают пари, и ректор обязуется в этом случае увеличить бюджет их факультета на следующий год на 5 процентов. Если ни один из них не угадает, бюджет заморозят на следующие 5 лет. Блефуют ли математики или у них есть возможность выиграть пари наверняка?

148. Свобода за одну монету

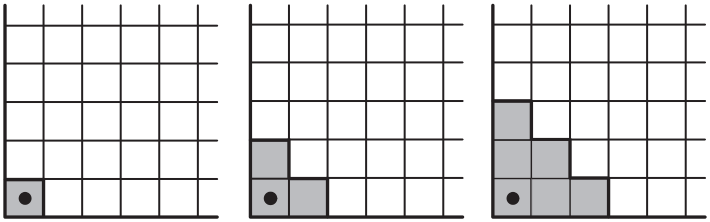
Тюремщик предлагает двум заключённым программистам — назовём их A и B — отпустить их, если они выиграют в следующей игре. Тюремщик ставит доску 8×8, на каждой клетке которой находится по монете; некоторые монеты лежат решками вверх, некоторые — решками вниз. Когда B выходит, тюремщик показывает программисту A клетку, которую B должен будет угадать. Заключённый A должен перевернуть ровно одну монету на доске и затем покинуть комнату. Потом заходит B и пытается угадать выбранную клетку. Заключённым разрешено спланировать их стратегию заранее, но запрещено общаться после начала игры. Разумеется, заключённому B разрешено смотреть на доску после того, как он войдёт в комнату, и проводить любые вычисления. Могут ли заключённые выиграть себе свободу, или в этой игре невозможно победить?

149. Распространение камушков

Рассмотрим игру для одного, в которую играют на бесконечном поле, образованном при делении первого квадранта плоскости на квадратные клетки. В начале игры один камешек ставится в угол поля. На каждом ходу игрок может взамен одного камешка поставить два на соседние клетки: один — непосредственно справа, а другой — непосредственно сверху, если обе эти клетки свободны. Цель игры — убрать все камешки из «лестницы» S_n, которая образована n последовательными диагоналями в углу поля (для примера см. рис. 2.34).

Например, при $n = 1$ первый и единственно возможный ход из начальной позиции освобождает S_1 (рис. 2.35).

Найдите все значения n, при которых возможно достичь цели игры.



(*а*) (*б*) (*в*)

Рис. 2.34. Начальная позиция в игре «*Распространение камушков*» для (*а*) $n = 1$, (*б*) $n = 2$ и (*в*) $n = 3$

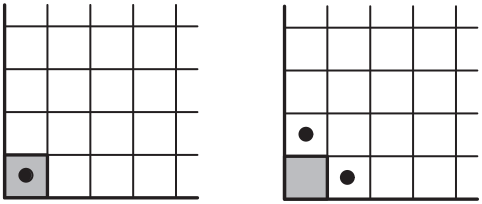


Рис. 2.35. Первый ход в игре «*Распространение камушков*» при $n = 1$ освобождает «лестницу» S_1 (отмечена серым)

150. Болгарский пасьянс

Возьмём n монет, где n — треугольное число (т. е. $n = 1 + 2 + \ldots + k$ для некоторого натурального k), и разделим

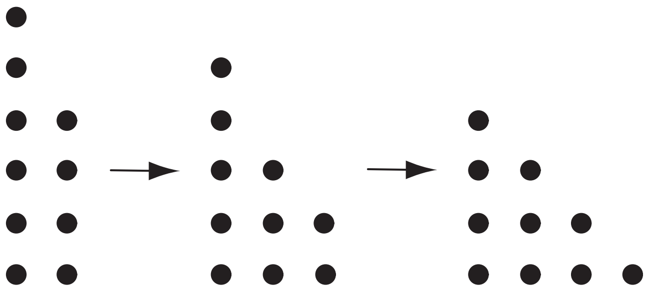


Рис. 2.36. Пример головоломки «*Болгарский пасьянс*»

их на $s \geqslant 1$ кучек, без ограничений на число кучек и на число монет в каждой кучке. Затем будем повторять следующую процедуру. Возьмём одну монету из каждой кучки и положим их все в новую кучку. Покажите, что вне зависимости от того, как изначально разделены n монет, такая процедура после конечного числа ходов всегда приведёт к k кучкам, содержащим $1, 2, \ldots, k$ монет соответственно. (Очевидно, что по достижении такого состояния алгоритм в нём и останется.) Например, рис. 2.36 иллюстрирует процесс для 10 монет, изначально разделённых на две кучки из 6 и 4 монет. Обратите внимание, что, поскольку для болгарского пасьянса порядок кучек не имеет значения, удобно раскладывать кучки в порядке невозрастания их размера, как на рис. 2.36.

Глава 3

ПОДСКАЗКИ

1. Волк, коза и капуста. За одним несущественным исключением, головоломку можно решить, делая последовательно шаги, которые единственно возможны в каждой ситуации.

2. Выбор перчаток. Представьте себе злобного соперника, который хочет, чтобы вы вытащили как можно больше перчаток перед тем, как достанете нужные. Имейте в виду, что перчатки — это не носки. Одни перчатки — на правую руку, другие — на левую.

3. Разделение прямоугольника. Не требуется, чтобы нужные треугольники были одного размера.

4. Отряд солдат. Сначала решите задачу, как переправить одного солдата.

5. Перестановки строк и столбцов. Ответ «нет». Определите почему.

6. Счёт на пальцах. Проиграйте достаточно долго, как девочка считает, и тогда ответ станет очевидным.

7. Переход по мосту ночью. Ответ «да», и решение не подразумевает никаких трюков.

8. Как собрать пазл? Похожая задача обсуждалась во второй части учебного раздела.

9. Счёт в уме. Существует по крайней мере два разных способа подсчитать сумму. Оба они используют методы, которые обсуждались во второй части учебного раздела.

10. Фальшивая монета из восьми. «Три» — ответ неверный.

11. Столбик фальшивых монет. Ответ «одно взвешивание». Не забудьте, что весы показывают точный вес.

12. Можно ли замостить доску? Ответ «нет».

13. Преграждённые пути. Примените динамическое программирование, как объяснялось в первой части учебного раздела.

14. Переделать шахматную доску. Какие части доски нужно разрезать, чтобы решить головоломку?

15. Замостить доску плитками тримино. Только на один из трёх вопросов ответ «да».

16. Печём блины. Как быстрее всего испечь три блина? Обратите внимание, что $n = 1$ — это случай особый.

17. Куда дойдёт король? Условиями головоломки королю не запрещено побывать на одной и той же клетке более одного раза. Кроме того, убедитесь, что ваш ответ верен для всех значений $n \geqslant 1$.

18. Проход из угла в угол шахматной доски. Обратите внимание на цвета клеток, на которые ходит конь.

19. Нумерация страниц. Найдите формулу, которая выражает общее количество знаков как функцию от количества страниц.

20. Спуск с максимальной суммой. Используйте стратегию динамического программирования.

21. Разбиение квадрата. Решения не существует всего для нескольких значений n. Обратите внимание, что более мелкие квадраты не обязательно должны быть одного размера.

22. Упорядочение списка команд. Упорядочить легко, если воспользоваться одной из стратегий разработки алгоритмов, описанной в первой части учебного раздела.

23. Задача о польском национальном флаге. Попробуйте переставлять по две шашки за раз.

24. Раскрашивание шахматной доски. Для каждой фигуры, кроме ладьи, решение можно найти непосредственным применением «жадной» стратегии. Простое решение для ладьи также найти нетрудно.

25. Лучшее время для жизни. Используйте указатель, данный в алфавитном порядке.

26. Тьюринг в списке. Может быть, будет легче найти количество «слов», который идут после TURING.

27. Игра «Икосиан». Не забудьте, что ваш путь не обязательно должен проходить по всем рёбрам, он должен проходить через все вершины. Можно попробовать перебор с возвратом. Если удача улыбнётся не сразу, запаситесь терпением.

28. Обвести фигуру. Задача о кёнигсбергских мостах, о которой говорилось во второй части учебного раздела, основана на том же принципе, что и данная головоломка.

29. Ещё раз о магическом квадрате. Обратитесь к первой части учебного раздела, там рассказывается о построении магических квадратов.

30. Ломание палки. Подумайте, как разломать самую длинную оставшуюся часть.

31. Трюк с тремя стопками карт. Обозначьте карты в трёх стопках после первого раскладывания, например, $a1, a2, \ldots, a9; b1, b2, \ldots, b9; c1, c2, \ldots, c9$, и проследите алгоритм.

32. Турнир на выбывание. Начните отвечать на вопросы, когда n — степень 2.

33. Магия и псевдомагия. В таблице $n \times n$ есть $(n-2)^2$ квадратов размером 3×3. Начните с ответов на вопросы головоломки для таблицы 4×4.

34. Монеты на звезде. Головоломку можно решить с помощью «жадной» стратегии или «метода пуговиц и верёвочек», описанного в первой части учебного раздела по разработке алгоритмов.

35. Три кувшина. Головоломку можно решить за шесть шагов.

36. Ограниченное разнообразие. Решите головоломку для $n = 2$, 3 и 4, и вы поймёте принцип.

37. Задача о 2n шашках. Задача упомянута в первой части учебного раздела, где говорится о стратегии «разделяй и властвуй».

38. Замощение плитками тетрамино. Ответ на четыре вопроса «да».

39. Прогулки по клеточному полю. Путь можно найти только для одной из досок, для другой — нет.

40. Перестановка четырёх коней. Классическая версия этой головоломки обсуждалась в первой части учебного раздела.

41. Круг света. Для разных n решение не одно и тоже. Рассмотрите несколько вариантов для малых значений n и вы увидите разницу.

42. Ещё раз о волке, козе и капусте. Сначала решите головоломку для $n = 1$, чтобы увидеть все возможные варианты расположения шашек.

43. Расстановка чисел. Начните с сортировки заданных чисел.

44. Легче или тяжелее? От вас не требуется найти фальшивую монету, только определить, легче она или тяжелее других.

45. Самый короткий путь коня. Минимальная последовательность ходов здесь достаточно очевидна. Доказать её оптимальность уже труднее. Но задача становится проще, если найти правильный способ измерить расстояние от начальной до конечной клетки.

46. Фишки трёх цветов. Решение есть для любых $n \geqslant 1$.

47. Планировка выставки. Ответ на первый вопрос практически очевиден. Ответ на второй вопрос возникает из стандартного применения идеи инварианта, которая обсуждалась во второй части учебного раздела.

48. Макнаггет-числа. Есть всего шесть целых чисел, которые не являются «макнаггет-числами». Для остальных чисел алгоритм основан на стратегии «уменьшай и властвуй».

49. Миссионеры и каннибалы. Решение головоломки — 11 переправ через реку. Похожая задача обсуждалась в первой части учебного раздела.

50. Последний шар. Подумайте о чётности.

51. Недостающее число. После того как вы «нащупали» ответ, постарайтесь улучшить его, чтобы Джилл смогла решить задачу в уме.

52. Подсчёт треугольников. Обратите внимание на число маленьких треугольников, добавляемых на каждой итерации. Похожий пример обсуждался в первой части учебного раздела.

53. Определение фальшивой монеты с помощью электронных весов. Какую информацию даёт взвешивание части монеток?

54. Разрезание прямоугольника. Может быть, лучше начать с головоломки «*Ломание палки*» (№ 30), которая является одномерной версией данной головоломки.

55. Головоломка «Одометр». На первый вопрос можно ответить с помощью стандартной комбинаторики. Если додумаетесь, и на второй можно ответить без громоздких расчётов.

56. Строй новобранцев. На первый вопрос ответ положительный. Конечно же, имелось в виду выстроить новобранцев иначе, но приказ ведь можно выполнить двумя способами.

57. Задача Фибоначчи о кроликах. Найдите уравнение, которое выражает зависимость числа кроликов после n месяцев от числа кроликов в какие-нибудь предыдущие месяцы.

58. Сортируем раз, сортируем два.... Для понимания головоломки решите задачу для небольшого двумерного массива карт или чисел.

59. Шапки двух цветов. Предположим, что только одна шапка чёрная. Как может заключённый, на которого она надета, догадаться об этом? А другие заключённые? Поняв это, вы сможете обобщить ответ и решить головоломку.

60. Переделка треугольника из монеток в квадрат. Для решения задачи нужно использовать формулу суммы первых n нечётных чисел: $S_n = 1 + 3 + \ldots + (2n - 1) = n^2$. Можно также по-другому визуализировать данный треугольник из монеток — чтобы вершина его прямого угла была вверху, а гипотенуза горизонтальна.

61. Шашки на диагонали. Можно решить головоломку для нескольких малых значений n, но лучший подход — это найти инвариант, т. е. характеристику, которая остаётся постоянной от шага к шагу.

62. Робот собирает монетки. Лучшая стратегия для этой задачи — динамическое программирование.

63. Плюсы и минусы. Используйте формулу $1 + 2 + \ldots + n = \frac{n(n+1)}{2}$ и исследуйте чётность суммы.

64. Восьмиугольники. Решите задачу сначала для восьми точек.

65. Угадывание кода. Можно придумать последовательность n-битовых строк, которая позволит на каждом шаге определять один новый бит в коде. Ответ легко описать.

66. Оставшееся число. Подумайте о чётности.

67. Разливаем пополам. Можете использовать жадную стратегию, но вам придётся доказать, что она ведёт к цели.

68. Сумма цифр. Возможно, легче решить задачу для общего случая, найдя общую сумму цифр во всех целых числах от 1 до $10n$, где n — натуральное число. Есть по меньшей мере три различных способа решить задачу для общего случая.

69. Фишки на секторах круга. Подумайте о чётности.

70. Прыжки в пары — 1. Чтобы найти минимальное количество монет, для которого задача имеет решение, можно использовать поиск с возвратом.

71. Помеченные ячейки — 1. Существует шесть значений n, при которых задачу решить нельзя; три из них очевидны.

72. Помеченные ячейки — 2. Ответы для чётных и нечётных значений n различны.

73. Погоня за петухом. Как фермер может ускорить поимку петуха, если петух старается убежать? Какой алгоритм позволяет сделать это как можно быстрее?

74. Выбор места. Для этой задачи есть гораздо более эффективный алгоритм, чем тот, который описан в учебном разделе. Решите сначала задачу для нескольких простых частных случаев, когда все дома расположены на одной улице, и это приведёт вас к нахождению алгоритма для общего случая.

75. Инспекция бензоколонок. Можно рассмотреть отдельно случаи для чётных и нечётных n.

76. Быстрая ладья. Оптимальный путь можно найти с помощью жадной стратегии. Однако доказать оптимальность не так легко.

77. Поиск закономерности. Ответы для десятичных и бинарных чисел различны.

78. Замощение прямыми тримино. Ответ «да».

79. Дверцы шкафчиков. Для ответа на вопросы примените алгоритм вручную, например для $n = 10$, и посмотрите, что получается.

80. Прогулка принца. С помощью одного и того же алгоритма принц пройдёт по всем клеткам шахматной доски размером $n \times n$ в точности по одному разу при всех положительных значениях n. Обратите внимание, что в условии не требуется, чтобы принц вернулся, т. е. чтобы путь закончился на клетке, расположенной в одном ходе принца от начальной.

81. Ещё раз о задаче о знаменитости. Более простая версия задачи решалась в первой части учебного раздела.

82. Вверх орлом. Используйте группы рядом расположенных орлов и решек.

83. Ханойская башня с ограничением. Головоломка решается с помощью рекурсивного алгоритма, как и её классическая версия (см. вторую часть учебного раздела).

84. Сортируем блины. Ваш алгоритм необязательно должен быть оптимальным, но во всяком случае он должен быть более эффективным, чем полный перебор.

85. Распространение сплетен — 1. Минимальное число сообщений для $n = 4$ — шесть.

86. Распространение сплетен — 2. Есть несколько алгоритмов, дающих ответ $2n - 4$ телефонных звонка при $n > 3$.

87. Перевёрнутые стаканы. Используйте чётность.

88. Жабы и лягушки. Головоломку можно решить с помощью последовательности ходов, которые достаточно очевидны, поскольку альтернативные ходы приводят в тупик. В Интернете также можно найти визуализацию решения этой головоломки.

89. Перестановка фишек. Определите, на какую головоломку похожа эта, и используйте похожий алгоритм.

90. Пересаживания. Есть простой алгоритм для получения всех перестановок путём транспозиции двух соседних элементов.

91. Горизонтальные и вертикальные домино. Нетривиальная часть задачи — доказать, что если n чётное, но не делится на 4, то замостить доску $n \times n$ равным числом горизонтальных и вертикальных плиток невозможно. Это можно сделать с помощью инварианта.

92. Замощение трапециями. Очевидное необходимое условие для того, чтобы замощение было возможно, также является и достаточным.

93. Стрельба по линкору. Пометьте минимальное количество клеток доски таким образом, чтобы любой прямоугольник размером 4×1 содержал бы по меньшей мере одну помеченную клетку.

94. Поиск в отсортированном массиве. Ответ «да».

95. Максимальный и минимальный вес. Для понимания задачи рассмотрите случай $n = 4$.

96. Замощение лестницы. Очевидного необходимого условия для замощения здесь не достаточно.

97. Обмен в колоде карт. Ответ «да». Игра всегда заканчивается после конечного числа итераций.

98. Ромбопалиндром. Для начала можно найти число способов прочитать CAT I SAW.

99. Обратная сортировка. Решите задачу для нескольких первых значений n и поймёте, как действовать дальше.

100. Куда доскачет конь. Определите форму области, содержащей клетки, до которых доскачет конь. Обратите внимание, что ответ для любых $n > 2$ даётся одной и той же формулой.

101. Перекраска комнат. Ответ «да».

102. Обезьянка и кокосовые орехи. Есть несколько остроумных способов решить головоломку. Но можно действовать и в лоб; нужно составить несколько уравнений и найти их наименьшее положительное целое решение.

103. Прыжки на другую сторону. Ответ «нет».

104. Разделение кучи фишек. Рассмотрите несколько простых случаев задачи.

105. Головоломка «MU». Ответ «нет».

106. Лампочка и переключатели. Попробуйте представить переключатели в виде битов (принимающих значения 0 и 1), хотя это и не обязательно.

107. Лиса и заяц. Лиса может поймать зайца в половине возможных значений s.

108. Самый длинный путь. Путь к решению поможет найти жадная стратегия.

109. Домино «дубль-n». На первые два вопроса можно ответить с помощью простых подсчётов. Кольцо можно построить для половины возможных значений n или с помощью рекурсии, или сведением к хорошо известной задаче о графах.

110. Хамелеоны. Посмотрите, как изменяется разность между количеством хамелеонов разных цветов после встречи двух хамелеонов.

111. Переворачивание треугольника из монеток. Найдите наилучший способ перевернуть треугольник, сделав k-й ряд монеток ($1 \leqslant k \leqslant n$) основанием перевёрнутого треугольника и определив затем оптимальное значение k.

112. Снова о покрытии домино. Ответ на вопрос прост. Указать способ замощения сложнее, потому что удалённые поля могут быть в любых местах на доске.

113. Исчезающие монеты. Решение нескольких простых частных случаев головоломки должно привести вас к правильной общей стратегии.

114. Обход точек. Решите головоломку для $n = 3$ и потом обобщите решение. Обратите внимание на то, что единственное ограничение для линий — это то, что они должны быть прямыми.

115. Задача Баше о гирях. Для решения обоих вариантов головоломки нетрудно догадаться применить жадный подход. Но суть задачи заключается в доказательстве оптимальности решения.

116. Подсчёт «пустых номеров». Ответ на первый вопрос можно получить, решив простое уравнение. Ответ на второй вытекает из ответа на первый.

117. Одномерный солитёр. Не считая симметричных решений, пустая клетка может быть расположена в двух местах. Соответственно, есть и два варианта расположения оставшегося колышка.

118. Шесть коней. Обратитесь к первой части учебного раздела, где обсуждается более простой вариант — головоломка Гуарини.

119. Замощение цветными тримино. Можно использовать стратегию, применённую в первой части учебного раздела для замощения той же доски нераскрашенными тримино.

120. Машина, распределяющая пенни. Пронумеруйте коробки слева направо начиная с 0 и представьте конечное распределение пенни с помощью битовой строки.

121. Проверка супер-яйца. Рассмотрите функцию $H(k)$, выражающую максимальное количество этажей, для которых задача может быть решена за k бросков.

122. Мир в парламенте. Начните с произвольного разделения парламентариев на две палаты, а затем найдите способ улучшить состав палат, как того требует задача.

123. Задача о флаге Нидерландов. Сначала решите головоломку «*Задача о польском национальном флаге*» (№ 23), где фишки только двух цветов.

124. Разделение цепочки. Начните с ответа на противоположный вопрос и найдите максимальную длину цепочки, которую можно получить, убрав k скрепок. Для $k = 1$ максимальная длина цепочки — семь скрепок.

125. Отсортировать 5 за 7. Хотя головоломку можно решить за семь взвешиваний (каждый раз сравнивая вес двух предметов), никакой общий алгоритм сортировки не даёт правильный подход к достижению этой цели.

126. Деление пирога по-честному. При $n = 2$ задача имеет простое, но остроумное решение, которое можно применить и для общего случая.

127. Задача о ходе коня. Есть очень много различных способов решения этой задачи, все они предлагают начинать с угловых полей. Делайте ходы конём так, чтобы он был как можно ближе к краю доски.

128. Тумблеры системы охраны. Хорошо помогает решение простых частных случаев задачи, а для общего случая можно применить стратегию «уменьшай и властвуй».

129. Головоломка Реве. Используйте тот же подход, что и для головоломки «*Ханойская башня*» (см., например, вторую часть учебного раздела).

130. Отравленное вино. Для части а) алгоритм может быть основан на нумерации бочек в двоичной системе. Ответ к части б) — 4.

131. Задача о шашках Тэта. Вам придётся решить головоломку для нескольких частных случаев, прежде чем вы найдёте нужную закономерность. В частности, решение для $n = 3$ направляет по неверному пути, а решение для $n = 4$ содержит необходимые компоненты для решения более сложных случаев. Общий случай можно решить с помощью алгоритма «уменьшай и властвуй», но сведение задачи к более простым частным случаям совсем неочевидно.

132. Солдаты Конвея. Воспользуйтесь решениями вариантов этой головоломки, которые вы уже знаете.

133. Игра «Жизнь». Минимальное число живых клеток для создания натюрмортов, осцилляторов и космических кораблей — 4, 3 и 5 соответственно.

134. Раскраска точек. Примените стратегию «уменьшай и властвуй».

135. Разные пары. Задачу можно решить или используя таблицу размером $2 \times n$, или разместив на окружности равноудалённые друг от друга точки.

136. Поимка шпиона. Сначала решите более простую задачу: вы знаете, что шпион начинает в момент времени 0, но вы можете задавать вопросы только начиная с момента времени 1.

137. Прыжки в пары — 2. Вспомните похожую задачу, и это поможет вам ответить на поставленный вопрос и разработать алгоритм.

138. Делёж конфет. Понаблюдайте, что происходит с самым большим и самым маленьким числом конфет, которое может быть у ребёнка.

139. Круглый Стол короля Артура. Примените стратегию итерационного улучшения, используя в качестве полуинварианта число пар врагов, сидящих рядом.

140. Снова задача об n ферзях. Рассмотрите отдельно шесть случаев различных остатков от деления n на 6. Случаи $n \bmod 6 = 2$ и $n \bmod 6 = 3$ сложнее, чем другие, и требуют корректировки жадного размещения ферзей.

141. Задача Иосифа Флавия. Выпишите два рекуррентных соотношения для местоположения выжившего $J(n)$: одно для чётных n и другое для нечётных n.

142. Двенадцать монет. Эту сложную задачу можно решить за три взвешивания, это как раз требуемый минимум.

143. Заражённая шахматная доска. Ответ n. Докажите, что это число является необходимым и достаточным для заражения всей доски.

144. Разрушение квадратов. Для поиска решения попробуйте замостить доску костяшками домино определённым образом. Минимальное количество спичек, которое нужно убрать с доски размером 4×4, — это 9.

145. Пятнашки. Найдите инвариант, который делает задачу при данном расположении невыполнимой.

146. Стрельба по движущейся мишени. Такой алгоритм существует. Пусть щитки пронумерованы от 1 до n; для начала рассмотрите случай, когда мишень находится за каким-либо щитком с чётным номером.

147. Шапки с номерами. Рассмотрите сумму чисел, которые видит каждый математик.

148. Свобода за одну монету. Игру можно выиграть применением стандартной вычислительной операции над битовыми строками.

149. Распространение камушков. Цели игры можно достичь только при $n = 1$ и $n = 2$. Найдите инвариант, который показывает, что ни для какого $n > 2$ это сделать невозможно.

150. Болгарский пасьянс. Сначала покажите, что алгоритм зацикливается при таком распределении монет, когда кучки формируются в порядке невозрастания их размера. Затем проследите передвижения каждой монетки в таком цикле, чтобы убедиться, что цикл содержит только одно распределение, соответствующее треугольному числу монет.

Глава 4

РЕШЕНИЯ

Авторы цитат в эпиграфе (в порядке цитирования) — Уильям Паундстоун, Дьёрдь Пойа, Мартин Гарднер, Карл Фридрих Гаусс и Фибоначчи.

ЛЁГКИЕ ГОЛОВОЛОМКИ

1. Волк, коза и капуста

Решение. Обозначим буквами M, w, g и c соответственно человека (крестьянина), волка, козу и кочан капусты. Человек всегда находится на том же берегу реки, где и лодка. Рисунок 4.1 показывает два варианта последовательности переправ через реку, которые дают решение.

Комментарии. Большинство головоломок не сводятся к таким простым решениям. Эта головоломка является редким исключением, поскольку у человека только один разумный выбор при каждой переправе, кроме третьей. Задачу можно также решить, используя граф пространства состояний (см. [Lev06, Section 6.6]), так же, как головоломку о двух ревнивых мужьях, о которой мы говорили в первой

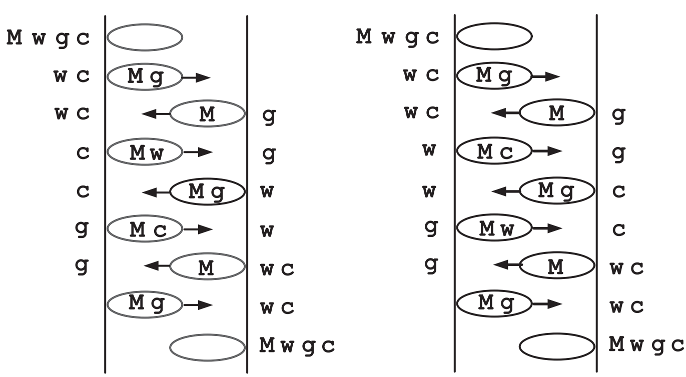


Рис. 4.1. Решение головоломки «*Волк, коза и капуста*»

части учебного раздела. Состояния задачи можно представить и как вершины куба (см., например, [Ste09, p. 256]). Различные способы решения дают очевидный ответ — минимум семь переправ. Эта классическая головоломка была включена в сборник Алкуина — самый ранний из известных сборников математических задач на латыни, о котором мы упоминали в первой части учебного раздела. О появлении головоломки в других частях света рассказано в [Ash90]. В наши дни эта задача очень часто включается в сборники головоломок (например, [Bal87, с. 118]; [Kor72, задача 11]). Удивительно, но она всё ещё привлекает внимание математиков и специалистов по информатике (см. [Cso08]).

2. Выбор перчаток

Решение. Ответ на вопросы а) и б) — 11 и 19 перчаток соответственно.

а) В худшем случае, до того как вы получите по крайней мере одну подходящую пару, вы вытащите 5 чёрных, 3 коричневых и 2 серых перчатки — все на одну руку. Следующая перчатка будет парной. Таким образом, ответ — 11 перчаток.

б) В худшем случае, до того как у вас будут пары перчаток каждого цвета, вы вытащите все 10 чёрных, все 6 коричневых и 2 серых на одну руку. Следующая серая перчатка будет парной. Таким образом, ответ — 19 перчаток.

Комментарии. Головоломка представляет собой простой пример анализа эффективности алгоритма в наихудшем случае.

В большинстве сборников головоломок приводится версия этой задачи для шаров разного цвета (например, [Gar78, p. 4–5]). Версия с перчатками, которая придаёт дополнительную особенность, использовалась в [Mos01, Problem 18].

3. Разделение прямоугольника

Решение. Прямоугольник можно разделить на n прямоугольных треугольников для любого целого числа $n > 1$.

При $n = 2$ прямоугольник разделяют, разрезая его вдоль диагонали (рис. 4.2, *а*). Если $n > 2$, то первый разрез делают вдоль диагонали, а затем проводят ещё $n - 2$ разрезов, деля любой из имеющихся прямоугольных треугольников на два прямоугольных треугольника. Для разделения прямоугольного треугольника на два прямоугольных треугольника

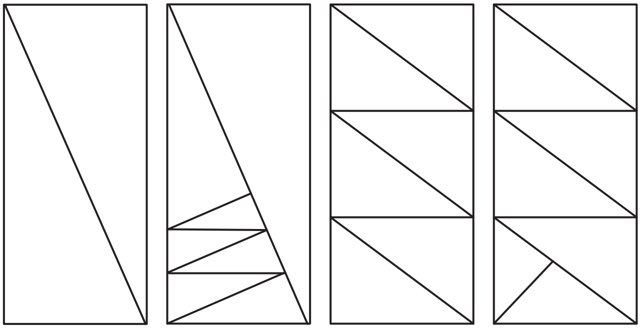


Рис. 4.2. Разделение прямоугольника на прямоугольные треугольники. Применение первого метода для (*а*) $n = 2$ и (*б*) $n=7$. Применение второго метода для (*в*) $n=6$ и (*г*) $n=7$

нужно провести разрез вдоль высоты треугольника, проведённой к гипотенузе. Пример показан на рис. 4.2, *б*.

Задачу можно также решить, сначала рассмотрев случай для чётного n: разрезать прямоугольник на $\frac{n}{2}$ меньших прямоугольников (например, с помощью $\frac{n}{2} - 1$ разрезов параллельно основанию прямоугольника) и затем каждый из них разрезать вдоль диагонали на два прямоугольных треугольника (рис. 4.2, *в*). Если n нечётное, мы сначала разрезаем этим же способом прямоугольник на $(n - 1)$ меньших треугольников, а затем разрезаем любой треугольник вдоль высоты, проведённой к гипотенузе (рис. 4.2, *г*).

Комментарии. Первое решение основано на «подходе по возрастающей» (стратегия «уменьшить на единицу», применённая в обратном направлении). Второе решение можно интерпретировать как пример стратегии «преобразуй и властвуй»: сведение нечётного случая к более простому чётному.

4. Отряд солдат

Решение. Сначала два мальчика переправляются в лодке на другую сторону, после чего один из них возвращается в лодке. Затем переправляется один солдат и остаётся на другом берегу, а мальчик возвращается в лодке. Эти четыре переправы уменьшают размер задачи, измеряемый количеством солдат, которым нужно переправиться, на единицу.

Итак, операция «четыре переправы» повторяется 25 раз, и задача будет решена за 100 переправ. (Разумеется, для общего случая n солдат нужно сделать $4n$ переправ.)

Комментарии. Эта лёгкая головоломка представляет собой хороший пример стратегии «уменьшай и властвуй» («уменьшай на единицу»). Данная стратегия разработки алгоритмов обсуждалась в первой части учебного раздела.

Эта старая и хорошо известная головоломка. Генри Э. Дьюдени опубликовал её в *Strand Magazine* в 1913 году (см. [Dud67, Problem 450]); она также была включена в русский сборник головоломок Е. И. Игнатьева [Ign78, задача 43], вышедший в 1908 году.

5. Перестановки строк и столбцов

Решение. Ответ «нет».

Перестановка строк сохраняет числа в строках, а перестановка столбцов сохраняет числа в столбцах. Для заданной таблицы (рис. 4.3) это не так. Например, 5 и 6 в исходной таблице находятся в одной строке, а в конечной — в разных.

Комментарии. Головоломка представляет собой хороший пример головоломки с инвариантом, который отличается от более известных инвариантов — чётности и раскраски.

Похожая головоломка включена в сборник А. Спивака [Spi02, задача 713].

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

→

12	10	11	9
16	14	5	13
8	6	7	15
4	2	3	1

Рис. 4.3. Исходная и конечная таблицы в головоломке «*Перестановки строк и столбцов*»

6. Счёт на пальцах

Решение. Девочка остановится на указательном пальце. Вот как начинается счёт на пальцах (Б — большой, У — указательный, С — средний, БЕ — безымянный, М — мизинец):

палец	Б	У	С	БЕ	М	БЕ	С	У
счет	1	2	3	4	5	6	7	8
счет	9	10	11	12	13	14	15	16
счет	17	18	19	20	21	22	23	24
счет	25	26	27	28	29	30	31	32

Легко видеть, что на один и тот же палец приходится каждое восьмое число. Поэтому для ответа на вопрос нужно найти остаток от деления 1000 на 8, который равен 0. Когда девочка досчитает до 1000, это число придётся на указательный палец (после среднего), так же, как и любое число, делящееся на 8.

Комментарии. Эта задача относится к довольно редкому типу алгоритмических головоломок, где нужно найти, что будет на выходе алгоритма (в данном случае алгоритм — процедура счёта на пальцах) в зависимости от заданного входа (число 1000). Головоломка взята из книги Мартина Гарднера [Gar06, задача 3.11]. Похожая задача есть в книге Генри Дьюдени [Dud67, задача 164].

7. Переход по мосту ночью

Решение. Последовательность шагов для решения головоломки показана на рис. 4.4.

В другом очевидном варианте решения человек 2 возвращается с фонариком после первого перехода, а человек 1 — после второго перехода.

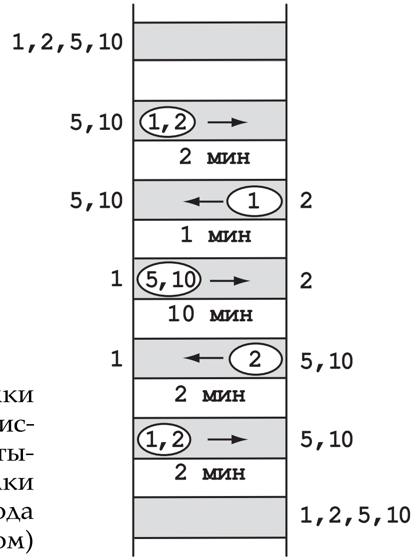


Рис. 4.4. Решение головоломки «*Переход по мосту ночью*». Числа 1, 2, 5, 10 обозначают четырёх людей соответственно. Стрелки показывают направление перехода (всегда с фонариком)

На самом деле 17 минут — минимальное необходимое время. Очевидно (и это можно формально доказать), что в оптимальном решении мост должны переходить два человека вместе и один должен вернуться с фонариком, пока все не окажутся на другом берегу. Таким образом, два человека должны сделать три перехода и один человек — два перехода; при этом все четверо окажутся на другом берегу за минимальное количество времени. Если фонарик оба раза возвращает самый быстрый человек, он должен быть в каждой паре, переправляющейся на другую сторону; общее время тогда будет $(10+1)+(5+1)+2 = 19$ минут. Если один раз из двух фонарик возвращает не самый быстрый человек, время возвращения с фонариком будет по меньшей мере $2+1 = 3$ минуты, а переходы на другую сторону по меньшей мере $10+2+2 = 14$ минут, поскольку по крайней мере в одной паре должен быть самый медленный человек (10 минут для прохода по мосту), а две другие пары перейдут минимум за 2 минуты каждая. Таким образом, всего понадобится 17 минут.

Комментарии. Как говорилось в первой части учебного раздела, эту головоломку нельзя правильно решить прямым применением жадного подхода. По этой причине многие считают задачу сложнее, чем она кажется на первый взгляд.

Головоломка, также известная как «Задача про мост и фонарик», широко обсуждалась в Интернете несколько лет назад. Уильям Паундстоун включил её в свою книгу как пример головоломки, решить которую предлагали во время интервью в *Microsoft* [Pou03, p. 86]. На веб-странице [Sillke] есть интересный материал по этой головоломке, в том числе самая ранняя ссылка на задачу в книге [Lev81] и алгоритм общего случая задачи, в котором n человек должны перейти по мосту с теми же ограничениями, как и в описанном нами варианте, и с произвольным временем перехода по мосту разными людьми. Доказательство оптимальности этого алгоритма было опубликовано Гюнтером Роте в 2002 году [Rot02]. Можно ещё обратиться к веб-сайту [Sni02] и статье [Bac08].

8. Как собрать пазл?

Решение. Ответ — 499 ходов.

Каждый ход уменьшает количество оставшихся секций на 1. Таким образом, после k ходов общее количество оставшихся секций будет $500 - k$, независимо от того, в каком порядке собираются секции. Итак, чтобы собрать весь пазл, потребуется 499 ходов.

Комментарии. Решение основано на той же идее инварианта, что и более известная головоломка «*Разделить плитку шоколада*» (см. вторую часть учебного раздела).

Головоломка была представлена Лео Мозером в *Mathematics Magazine* в январе 1953 года (p. 169). Позднее она была включена в [Ave00, Problem 9.22].

9. Счёт в уме

Решение. Сумма равна 1000.

Цель — подсчитать в уме сумму чисел в таблице на рис. 4.5.

Первый способ основан на наблюдении, что сумма любых двух чисел в клетках, симметричных относительно диагонали, исходящей от нижнего левого к верхнему правому углу, равна 20: $1 + 19$, $2 + 18$, $2 + 18$ и т. д. Поскольку есть $\frac{10 \cdot 10 - 10}{2} = 45$ таких пар (мы вычли число клеток, находящихся на самой диагонали, из общего числа клеток), сумма чисел вне диагонали равна $20 \cdot 45 = 900$. Сумма чисел на диагонали равна $10 \cdot 10 = 100$; общая сумма $900 + 100 = 1000$.

Второй способ — вычислить сумму по строкам (или по столбцам). Сумма в первой строке, как объяснялось во второй части учебного раздела, равна $\frac{10 \cdot 11}{2} = 55$. Сумма во второй строке равна 55+10, поскольку каждое число в ней на 1 больше, чем соответствующее число в первой строке. Это

1	2	3			⋯			9	10
2	3						9	10	11
3						9	10	11	
					9	10	11		
				9	10	11			
⋮			9	10	11				⋮
		9	10	11					
	9	10	11						17
9	10	11						17	18
10	11				⋯		17	18	19

Рис. 4.5. Таблица чисел, сумму которых нужно найти в головоломке «*Счёт в уме*»

относится и к другим строкам. Таким образом, общая сумма равна $55+(55+10)+(55+20)+\ldots+(55+90)=55\cdot10+(10+20+\ldots$ $\ldots+90)=55\cdot10+10\cdot(1+2+\ldots+9)=55\cdot10+10\cdot45=1000$.

Комментарии. В первом способе используется тот же трюк, который Карл Гаусс предположительно применил для нахождения суммы первых ста натуральных чисел, о чем рассказано во второй части учебного раздела. Мы говорили, что эта формула сама по себе очень важна для анализа алгоритмов. Мы использовали эту формулу дважды во втором способе решения данной головоломки.

Эта задача похожа на вопрос 1.33 в сборнике [Cra07].

10. Фальшивая монета из восьми

Решение. Ответ — два взвешивания.

Выберите две кучки по три монетки и положите их на разные чаши весов. Если они весят одинаково, то фальшивая монета — одна из двух оставшихся. Взвесьте их, и определите, какая легче (она и будет фальшивой). Если вес кучек монет при первом взвешивании разный, то фальшивая — в числе трёх более лёгких. Возьмите любые две из них и положите на разные чаши весов. Если они весят одинаково, то фальшивая — третья. Если они разного веса, то фальшивая — более лёгкая. Задачу нельзя решить одним взвешиванием. Алгоритм, предполагающий два взвешивания, является оптимальным.

Комментарии. Поскольку $8=2^3$ и уменьшение размера задачи вдвое обычно подсказывает очень эффективный алгоритм, понятно, что многие решают задачу за три взвешивания вместо двух. Однако данная головоломка представляет собой редкий пример задачи, размер которой можно уменьшить более чем в 2 раза. Головоломка также иллюстрирует особенность задач с конкретными числовыми данными. Иногда можно извлечь выгоду из каких-то особенностей данных, а иногда они вводят в заблуждение (как в данном примере).

У задачи есть другое решение, в котором второе взвешивание не зависит от результатов первого. Обозначим монеты буквами A, B, C, D, E, F, G, H. При первом взвешивании взвесим с одной стороны монеты A, B, C, а с другой — F, G, H. При втором взвешивании сравним вес A, D, F с весом C, E, H. Если $ABC = FGH$ (результаты первого взвешивания на весах), все шесть монет настоящие, и поэтому второе взвешивание эквивалентно взвешиванию D с E. Если $ABC < FGH$, только A, B и C могут быть фальшивыми. Поэтому, если при втором взвешивании $ADF = CEH$, то B фальшивая, если $ADF < CEH$, то A фальшивая, если $ADF > CEH$, то фальшивая C. Вариант, при котором $ABC > FGH$, является симметричным.

Решение головоломки нетрудно представить в общем случае с произвольным числом монет, хотя оптимальное доказательство алгоритма «разделение на три части» обычно проводят с помощью более продвинутых методов, таких как дерево решений (например, [Lev06, разд. 11.2]).

По словам Т. О'Бейрна [OBe65, p. 20], головоломка относится к периоду Первой мировой войны. В наши дни в Соединённых Штатах её часто предлагают решить во время интервью при приёме на работу. Гораздо более сложный вариант — с 12 монетами (№ 142).

11. Столбик фальшивых монет

Решение. Головоломку можно решить за одно взвешивание.

Пронумеруйте столбики монет от 1 до 10. Возьмите одну монетку из первого столбика, две из второго и т. д., все 10 — из последнего. Взвесьте все эти монеты вместе. Разница между весом этих монет и весом 550 граммов (столько должны весить $1 + 2 + \ldots + 10 = 55$ настоящих монет) и указывает на количество фальшивых монет, которое равно номеру столбика. Например, если выбранные монеты весят 553 грамма, то 3 монеты — фальшивые, и значит, фальшивый — третий столбик.

Комментарии. Решение основано на идее изменения представления. Эта задача была включена в первый сборник колонок Мартина Гарднера в *Scientific American* [Gar88a, с. 26] и в книгу [Ave00, Problem 9.11].

12. Можно ли замостить доску?

Решение. Замостить доску нельзя. Это можно доказать от противного. Допустим, что замостить можно. Поскольку доска симметрична, предположим, что её левая верхняя клетка покрыта горизонтальным домино под номером 1 на рис. 4.6. Тогда клетка в первом столбце второй строки сверху должна быть покрыта вертикальным домино, а клетка во втором

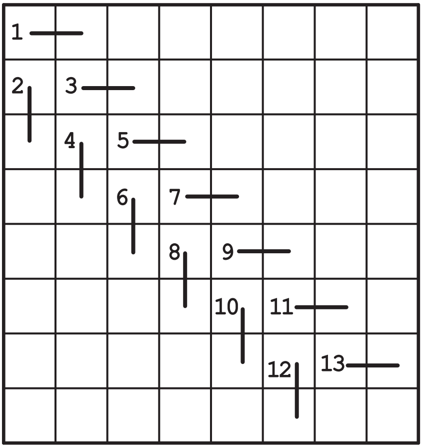


Рис. 4.6. Решение головоломки «*Можно ли замостить доску?*»

столбце этой же строки должна быть покрыта горизонтальным домино. Следуя той же логике, получаем замощение, показанное на рис. 4.6. Если после 13-го домино мы разместим домино горизонтально (а это единственно возможный вариант), то это будет противоречить условию, что никакие два домино не должны образовывать квадрат 2×2.

Комментарии. Это довольно редкий пример доказательства невозможности, которое не основано на той или иной идее инварианта, упоминаемой во второй части учебного раздела.

Данная головоломка — задача 102 из сборника [Fom96].

13. Преграждённые пути

Решение. Ответ — 17 путей.

Самый лёгкий способ решить задачу — применить динамическое программирование — одну из стратегий, описанных в первой части учебного раздела. Этот подход вычисляет количество кратчайших путей от точки A до любой точки пересечения линий вне ограждённой зоны (см. рис. 4.7). Начнём с присвоения 1 пересечению A и подсчитаем эти количества по строкам слева направо в каждой строке. Если у точки пересечения есть соседние точки и слева и сверху, число путей вычисляется как сумма этих соседних чисел. Если у точки пересечения только одна соседняя точка, то ей приписывается то же число, что и у соседней.

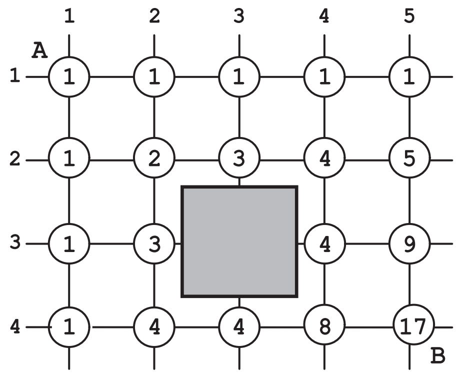


Рис. 4.7. Подсчёт кратчайших путей от A до B с ограждённым участком (показан серым)

Комментарии. Похожая задача была рассмотрена в первой части учебного раздела. Подсчёт путей — это широко известное применение динамического программирования (см., например, [Gar78, p. 9–11]). Другие случаи использования динамического программирования, как правило, сложнее.

14. Переделать шахматную доску

Решение. Ответ — 25 частей.

Поскольку на обычной шахматной доске нет ни одного кусочка размером 2×1 или 1×2, окрашенного в один и тот же цвет, каждую часть размером 4×4 нужно разрезать и по горизонтали, и по вертикали. Четыре разреза по горизонтали и четыре по вертикали, как показано на рис. 4.8, рассекают доску на 25 частей. Это и будет мини-

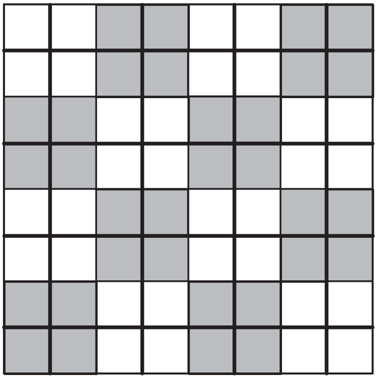


Рис. 4.8. Наилучший способ разрезать доску, чтобы переделать её в обычную шахматную

мальным: четыре квадрата 1×1, двенадцать прямоугольников 1×2 и девять квадратов 2×2. Все они раскрашены как обычная шахматная доска. Обычную шахматную доску можно собрать различными способами. Например, можно просто повернуть восемь прямоугольников 1×2 на краю доски на 180^0 и повернуть четыре квадрата 2×2 на 90^0.

Комментарии. Задача взята из книги Сергея Грабарчука [Gra05, p. 31].

15. Замостить доску плитками тримино

Решение. Ответ «нет» на вопросы а) и б) и «да» на вопрос в).

а) Ответ «нет», поскольку доску размером 3×3 нельзя замостить угловыми тримино. Угол доски (например, нижний левый) можно замостить одним тримино, расположив его тремя различными способами. При этом остаётся пространство только для ещё одного углового тримино (рис. 4.9).

б) Ответ «нет», поскольку общее число клеток любой доски размером $5^n \times 5^n$ на три не делится.

в) Замостить доску можно легко, разделив её на прямоугольники размером 2×3 и покрыв каждый из них двумя тримино (например, см. рис. 4.10).

Комментарии. Ответ на первый вопрос можно получить путём внимательного рассмотрения самой малой частной задачи. (Самая малая частная задача в данном случае является исключением, поскольку доски размером $3^n \times 3^n$, где $n > 1$, замостить тримино можно [Mar96, p. 31].) Ответ на второй

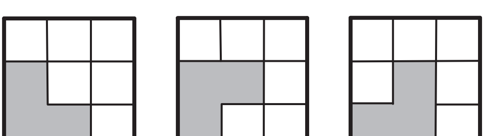


Рис. 4.9. Три варианта начать замощение доски размером 3×3, поместив тримино в нижний левый угол

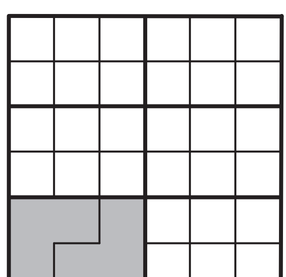


Рис. 4.10. Замощение доски размером 6×6

вопрос основан на идее инварианта. Решение третьего вопроса в итоге сводится к применению стратегии «разделяй и властвуй».

Головоломка сходна с задачей 50 в книге [Par95].

16. Печём блины

Решение. Минимальное время — n минут для каждого $n > 1$ и 2 минуты для $n = 1$.

Если n чётное, то решение очевидно: каждую пару блинов нужно печь одновременно, сначала с одной стороны, затем с другой.

Если $n = 1$, то нужно 2 минуты, чтобы испечь блин с обеих сторон. Если $n = 3$, то нужно 3 минуты. Сначала печём блины 1 и 2 с одной стороны. Затем печём блин 1 со второй стороны вместе с блином 3 с первой стороны. Потом печём блины 2 и 3 со второй стороны. Если n нечётное и больше 3, оптимальный алгоритм такой: сначала испечь таким способом первые три блина, и затем оставшиеся $n - 3$ (чётное число), как описано выше.

Для каждого $n > 1$ такому алгоритму требуется n минут. Это минимально возможное время, поскольку у n блинов $2n$ стороны, и никакой алгоритм не поможет испечь больше двух сторон за 1 минуту.

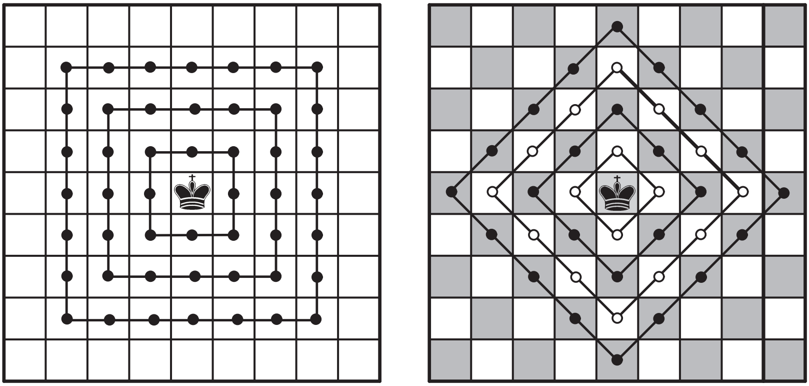
Комментарии. Этот алгоритм можно рассматривать как алгоритм «уменьшение на два». Но ключом к пониманию задачи является оптимальный способ испечь три блина.

Согласно библиографии Дэвида Сингмастера [Sin10, разд. 5.W], самое раннее упоминание головоломки относится к 1943 году, хотя он говорит, что головоломка, скорее всего, была известна ранее. После этого она была включена во множество сборников головоломок (например, [Gar61, p. 96]; [Bos07, с. 9, задача 38]).

17. Куда дойдёт король?

Решение. а) Ответ $(2n+1)^2$ для $n > 1$ и 8 для $n = 1$.

За один ход король может дойти до любого из восьми полей, прилегающих к исходному. После двух ходов он может быть либо на исходном поле (уйдя с него и вернувшись), либо на любом из 8 полей, примыкающих к исходному (перейдя на соседнее поле и затем на одно из ближайших восьми полей), либо на любом из 16 полей, образующих средний квадрат, показанный соединёнными точками на рис. 4.11, *а*.



(*а*) (*б*)

Рис. 4.11. До каких полей может дойти король (*а*) после трёх ходов, если будет ходить как обычный шахматный король и (*б*) после трёх ходов по горизонтали или вертикали (белые кружочки) и после четырёх ходов по горизонтали или вертикали (чёрные кружочки) (исходное поле обозначено в центре)

Таким образом, все поля, достижимые за два хода, находятся на периметре этого квадрата или же внутри него. В общем случае после $n > 1$ ходов король может дойти только до полей, находящихся по периметру квадрата $(2n+1) \times (2n+1)$ или внутри него с центром в исходном поле (см. рис. 4.11, *а* для $n = 3$). Количество таких полей равно $(2n+1)^2$. Если $n = 1$, король может дойти до восьми полей, прилегающих к исходному. Но, в отличие от случая $n > 1$, он не может вернуться на исходное поле.

б) Ответ $(n+1)^2$. Если король ходит только по горизонтали или вертикали, через n ходов он всегда будет стоять на поле того же цвета, что исходное, если n чётное, или же на поле противоположного цвета, если n нечётное. Рассмотрим самые дальние поля, до которых король может дойти за n ходов. Эти поля формируют границу: все поля на этой границе и все поля того же цвета внутри границы достижимы для короля за n ходов (см. рис. 4.11, *б*). Таких полей существует $(n+1)^2$.

Комментарии. Верность такого решения можно доказать боле строго с помощью математической индукции. Далее в книге есть сходная задача для шахматного коня — головоломка «*Куда доскачет конь*» (№ 100).

18. Проход из угла в угол шахматной доски

Решение. Такой проход невозможен. Поля начала и конца одного хода коня всегда противоположного цвета. Чтобы пройти по всем полям доски по разу, конь должен сделать 63 хода. Поскольку это число нечётное, такой проход должен начинаться и заканчиваться на полях противоположного цвета. Но цвет нижнего левого и верхнего правого полей одинаковый, поэтому проход невозможен.

Комментарии. Эта головоломка — стандартный пример использования шахматной раскраски доски для поиска инварианта. Обратите внимание, что у задачи о нахождении маршрута коня по всем полям стандартной шахматной доски размером 8×8 есть решение в случае, если не требуется, чтобы маршрут начинался и заканчивался на противоположных по диагонали полях. Эта задача известна как «*Задача о ходе коня*» (№ 127).

19. Нумерация страниц

Решение. Ответ — 562 страницы.

Пусть $D(n)$ — это общее количество знаков в первых n натуральных числах (номерах страниц). В первых девяти числах по одному знаку. Таким образом, $D(n) = n$ при $1 \leqslant n \leqslant 9$. В следующих 90 числах, от 10 до 99 включительно, по два знака. Поэтому

$$D(n) = 9 + 2(n - 9) \quad \text{при} \quad 10 \leqslant n \leqslant 99.$$

Максимальное значение $D(n)$ для этого диапазона $D(99) = 189$. Это означает, что нужно ещё какое-то количество трёхзначных чисел, чтобы общее количество знаков равнялось 1578 по условиям головоломки. Трёхзначных чисел существует 900, что даёт формулу

$$D(n) = 189 + 3(n - 99) \quad \text{при} \quad 100 \leqslant n \leqslant 999.$$

Чтобы найти ответ, нужно решить уравнение

$$189 + 3(n - 99) = 1578.$$

Комментарии. Эта головоломка включена в книгу в качестве примера простого алгоритмического анализа. Похожие задачи часто встречаются в несложных книгах по занимательной математике.

20. Спуск с максимальной суммой

Решение. С помощью стандартного метода динамического программирования, о котором говорилось в первой части учебного раздела, подсчитайте максимальную сумму вдоль спуска от вершины до каждого числа в треугольнике. Начните с вершины, где сумма равна самому числу. Затем рассчитайте суммы, двигаясь сверху вниз и, например, слева направо вдоль рядов чисел треугольника следующим образом. Для каждого первого и последнего числа в ряду сложите сумму, рассчитанную для соседнего числа в предыдущем ряду, с самим числом. Для каждого числа, которое не является первым или последним в ряду, сложите наибольшую сумму из рассчитанных для двух соседних чисел в предыдущем ряду, с самим числом. Когда все суммы подсчитаны для чисел в основании треугольника, найдите максимальную из них.

Рисунок 4.12 иллюстрирует применение алгоритма для заданного треугольника.

Комментарии. Головоломка взята с веб-сайта «Проект Эйлер» [ProjEuler].

```
      2                  2
    5   4              7   6
  3   4   7         10  11  13
1   6   9   6     11  17  22  19

     (а)                (б)
```

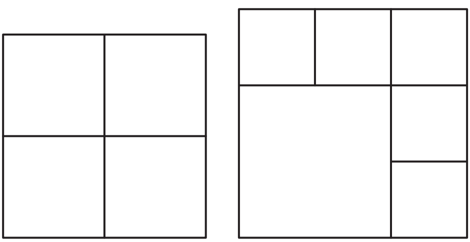
Рис. 4.12. Иллюстрация применения алгоритма динамического программирования для решения головоломки «*Спуск с максимальной суммой*». (*а*) Исходный треугольник. (*б*) Треугольник из максимальных сумм, рассчитанных по нисходящим путям; максимальная сумма равна 22

21. Разбиение квадрата

Решение. Квадрат можно разбить на n более мелких квадратов для каждого $n > 1$, кроме $n = 2$, 3 и 5.

Учитывая то, что четыре прямых угла данного квадрата должны находится в более мелких квадратах, ясно, что для этих трёх значений n решения у задачи нет. Для $n = 4$ есть одно очевидное решение, которое показано на рис. 4.13, *а*. Это решение можно обобщить на случай любого чётного $n = 2k$, расположив $2k - 1$ равных квадратов вдоль двух смежных сторон данного квадрата. Длина стороны каждого маленького квадрата равна $\frac{1}{k}$ от длины стороны большого квадрата. Рисунок 4.13, *б* показывает такое решение для $n = 6$.

Если $n > 5$ и является нечётным, т. е. $n = 2k + 1$, где $k > 2$, тогда $n = 2(k - 1) + 3$. Сначала мы можем разбить данный квадрат на $2(k - 1)$ квадратов, как описано выше, а затем разбить какой-нибудь из полученных квадратов (например, расположенный в верхнем левом углу) на четыре более мелких, что увеличит общее количество полученных квадратов на 3. Такое решение для $n = 9$ показано на рис. 4.14.



(*а*) (*б*)

Рис. 4.13. Разбиение квадрата на (*а*) четыре квадрата и (*б*) шесть квадратов

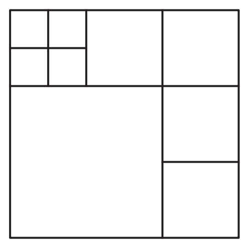


Рис. 4.14. Разбиение квадрата на девять квадратов

Комментарии. Мы рассмотрели данную задачу отдельно для случаев чётного и нечётного n и свели более сложный случай (для нечётных n) к более простому (для чётных n).

Головоломка была включена в несколько сборников (например, [Sch04, p. 9–11]). Разумеется, задача о разбиении квадрата на квадраты разных размеров гораздо сложнее (история этой задачи хорошо описана в [Ste04, Chapter 13]).

22. Упорядочение списка команд

Решение. Задача решается с помощью рекурсивного алгоритма. Если $n = 1$, задача решена. Если $n > 1$, решим задачу рекурсивно для произвольно выбранной группы из $n - 1$ команд. Затем просмотрим полученный список из этих $n - 1$ команд и добавим команду, не включённую в группу, сразу перед первой командой из списка, которая проиграла ей в турнире. Если такой команды нет, значит, команда, которой ещё нет в списке, проиграла всем командам из списка; тогда вставим её в конец списка.

Комментарии. Этот алгоритм является идеальной иллюстрацией стратегии «уменьшай и властвуй». Его также можно применять в обратной последовательности. Начнём упорядочение, поставив команду 1 на первое место, а затем последовательно будем вставлять в список команды $2, 3, \ldots, n$ непосредственно перед первой командой в списке, которая проиграла им в турнире. Если такой команды нет, т. е. если команда, которой ещё нет в списке, проиграла все игры с командами в списке, то поместим эту команду в конец списка.

Эта головоломка известна давно. Например, её версия о шахматном турнире (а партия в шахматы может закончиться вничью) была описана в книге Е. Гика [Gik76, с. 179].

23. Задача о польском национальном флаге

Решение. Вот один из алгоритмов для решения задачи. Найдём самую левую из белых шашек и самую правую из красных. Если самая левая белая шашка находится правее от самой правой красной, задача решена. Если нет, поменяем их местами и повторим операцию.

Рисунок 4.15 иллюстрирует этот алгоритм.

Комментарии. Этот алгоритм похож на основную часть алгоритма быстрой сортировки *quicksort*, одного из основных алгоритмов сортировки (см., например [Lev06, разд. 4.2]). Его можно рассматривать как стратегию «уменьшай и властвуй», в которой степень уменьшения размера задачи может меняться от одной итерации к другой.

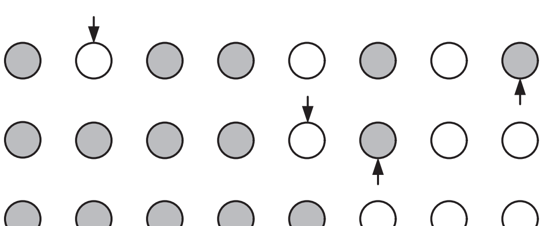


Рис. 4.15. Иллюстрация алгоритма для решения задачи о польском национальном флаге

Головоломка является упрощённой версией головоломки «*Задача о флаге Нидерландов*» (№ 123), которая описана далее в книге.

24. Раскрашивание шахматной доски

Решение. а) Для коня минимальное число цветов при $n > 2$ — два. Очевидно, что нужно более одного цвета, и стандартная раскраска доски в два противоположных цвета как раз и даёт ответ на первый вопрос. При $n = 2$ ответ — один: два коня на такой маленькой доске не могут угрожать друг другу.

б) Поскольку слон угрожает всем клеткам на одной с ним диагонали и никаким другим, по крайней мере нужно n цветов, чтобы раскрасить главную диагональ доски от верхнего левого до нижнего правого угла. Самый простой способ раскрасить все доску — раскрасить все клетки в каждом ряду по вертикали в тот же цвет, что и клетка на главной диагонали. Итак, для слона ответ — n.

в) Так как король угрожает только каждой соседней с ним клетке по горизонтали, вертикали или диагонали, нужно по крайней мере четыре цвета, чтобы закрасить каждую часть доски размером 2×2. Разделим доску на такие непересекающиеся участки (некоторые из них могут быть прямоугольниками меньшего размера и рассматриваться как участки размером 2×2 с некоторыми клетками, выходящими за края доски) и раскрасим каждый участок 2×2 четырьмя цветами по одной схеме. Итак, для короля получаем ответ четыре.

г) Поскольку ладья угрожает только всем клеткам в своём ряду по горизонтали или вертикали, необходимо по крайней мере n цветов, чтобы раскрасить каждый ряд (горизонтальный или вертикальный). Это число является не только необходимым, но и достаточным. Самый простой способ раскрасить доску в n цветов так, чтобы никакие две клетки в одном ряду не были одного цвета, — это раскрасить, например, первый горизонтальный ряд в n цветов, а затем смещать цветовую схему направо на один вертикальный ряд, окрашивая цветами, которые выходят за пределы доски, клетки в самых левых вертикальных рядах.

Пример для $n = 5$ показан на рис. 4.16.

Комментарии. Прямые решения для коня, слона и короля можно интерпретировать как применение жадной стратегии. Что касается ладьи, решение представляет собой *латинский*

1	2	3	4	5
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1

Рис. 4.16. Раскрашивание доски размером 5×5 пятью цветами таким образом, чтобы никакие две клетки в одном ряду по горизонтали или вертикали не были одного цвета

квадрат n-го порядка — таблицу размером $n \times n$, заполненную различными n символами таким образом, чтобы каждый символ встречался в точности один раз в каждом ряду по вертикали и в точности один раз в каждом ряду по горизонтали. Для коня, слона и короля ответ на вопрос о раскрашивании доски лёгок, а для ферзя это не так (см. [Iye66]).

25. Лучшее время для жизни

Решение. В общих терминах задачу можно сформулировать следующим образом. Даны n интервалов $(b_1, d_1), \ldots, (b_n, d_n)$, где в нашем случае b_i, d_i — это даты рождения и смерти i-го человека в указателе $(1 \leqslant i \leqslant n)$. Нужно найти интервал, который является пересечением самого большого количества заданных интервалов. Все интервалы являются открытыми, т. е. не включают в себя конечные точки. Если $d_i = b_j$, закрывающие скобки i-го интервала предшествуют открывающим скобкам j-го интервала. Если у нескольких интервалов открывающие скобки одни и те же, то их нужно считать для каждого из этих интервалов. Разумеется, это же касается и совпадающих закрывающих скобок.

Полезно представить интервалы нарисованными на вещественной оси, как на рис. 4.17. Последовательность скобок интервалов даёт ключ к эффективному решению задачи. Нетрудно увидеть, что задачу можно решить, просмотрев последовательность слева направо и подсчитав количество скобок (+1 для каждой открывающей скобки и −1 для закрывающей).

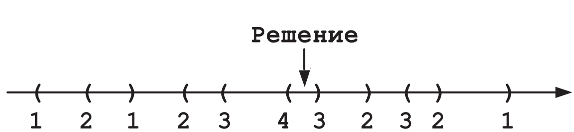


Рис. 4.17. Пояснение алгоритма решения головоломки «*Лучшее время для жизни*»

На левой скобке искомого интервала счёт достигает максимума, а правая скобка — это просто ближайшая к ней.

Комментарии. Представление входных данных как интервалов на вещественной оси — это пример изменения представления в стратегии «преобразуй и властвуй», описанной в первой части учебного раздела.

26. Тьюринг в списке

Решение. Ответ 598.

Общее количество слов, составленных из шести букв, равно $6! = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 720$ (см. первую часть учебного раздела). «Слова», следующие в списке за словом TURING в алфавитном порядке, имеют форму или U$*****$ или TURN$**$, где $*$ — это одна из шести букв, которой ещё нет в «слове». Поскольку эти буквы могут располагаться в любом порядке, то «слов» насчитывается $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ и $2! = 2 \cdot 1$ соответственно (см. первую часть учебного раздела). Таким образом, общее количество «слов», следующих за словом TURING в алфавитном порядке, равно $5! + 2! = 120 + 2 = 122$. Это означает, что TURING будет в списке на $720 - 122 = 598$ месте, если пронумеровать элементы списка от 1 до 720.

Комментарии. Головоломка представляет собой частный случай хорошо известной задачи об упорядочении перестановок. Алгоритм «уменьшить на единицу» для её решения описан, например, в [Kre99, p. 54–55].

27. Игра «Икосиан»

Решение. У головоломки 30 решений, одно из которых показано на рис. 4.18.

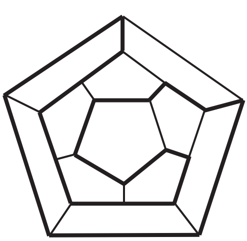


Рис. 4.18. Одно из решений головоломки «*Игра „Икосиан“*»

Комментарии. Головоломка является частным случаем одной из наиболее захватывающих задач о графах. Это задача о существовании гамильтонова цикла — последовательности смежных (соединённых ребром) вершин, которая начинается в какой-либо вершине, проходит все другие вершины в точности по одному разу, а затем возвращается к исходной вершине.

Для некоторых графов гамильтонов цикл существует — как, например, для графа игры «Икосиан», а для других нет. Эффективный алгоритм, который позволял бы определить, существует ли гамильтонов цикл для произвольного графа, неизвестен. Большинство специалистов по информатике считают, что его не существует. Несмотря на более чем 50-летние поиски доказательства этой гипотезы и премию в 1 миллион долларов, объявленную в 2000 году за решение этого вопроса, он остаётся нерешённым.

28. Обвести фигуру

Решение. Анализ задачи о кёнигсбергских мостах из второй части учебного раздела показывает, что фигуру можно обвести, не отрывая ручки от бумаги и не проходя обратно по какой-либо линии этой фигуры, тогда и только тогда, когда мультиграф фигуры является связным и удовлетворяет одному из двух условий:

- Все вершины мультиграфа имеют чётные степени (т. е. чётное число рёбер, для которых данная вершина является конечной точкой). Тогда начать обводить фигуру можно в любой вершине, на ней же и закончить.
- В точности две вершины имеют нечётную степень. Тогда нужно начать обводить с одной из этих нечётных вершин и закончить на другой.

а) Первую фигуру обвести с учётом накладываемых ограничений можно: её граф (рис. 4.19, *а*) является связным и все его вершины имеют чётные степени.

Для построения эйлерова цикла есть хорошо известный алгоритм. Путь начинается в произвольно выбранной вершине и идёт вдоль ещё не пройденных рёбер до тех пор,

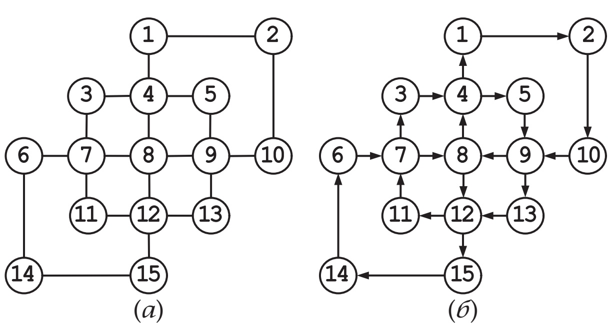


Рис. 4.19. (а) Граф фигуры, которую нужно обвести. (б) Эйлеров цикл графа

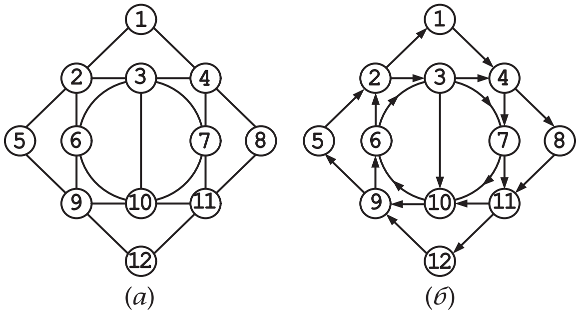


Рис. 4.20. (а) Граф фигуры, которую нужно обвести. (б) Эйлеров цикл графа

пока не пройдёт по всем рёбрам или же вернётся в исходную вершину, не пройдя по некоторым рёбрам ввиду невозможности сделать это. В последнем случае пройденный цикл «изымается» из графа и та же операция повторяется рекурсивно, начиная с вершины, которая находится как в оставшейся части графа, так и в изъятом цикле. (Существование такой вершины следует из связности графа и из чётности степеней всех его вершин.) После того, как эйлеров цикл построен для оставшейся части графа, он «сращивается» с первым циклом. Тем самым мы получаем эйлеров цикл для всего графа.

Например, начав с вершины 1 графа на рис. 4.19, *а* и проходя по «внешним» рёбрам, мы получаем такой цикл:

$$1 - 2 - 10 - 9 - 13 - 12 - 15 - 14 - 6 - 7 - 3 - 4 - 1.$$

Выбрав, например, вершину 4 в качестве общей с оставшейся частью графа, получим следующий эйлеров цикл для оставшейся части графа:

$$4 - 5 - 9 - 8 - 12 - 11 - 7 - 8 - 4.$$

«Сращивая» второй цикл с первым, получаем эйлеров цикл для всего графа (рис. 4.19, *б*):

$$1 - 2 - 10 - 9 - 13 - 12 - 15 - 14 - 6 - 7 - 3 - 4 - 5 - 9-$$
$$8 - 12 - 11 - 7 - 8 - 4 - 1.$$

б) Вторую фигуру тоже можно обвести с учётом наложенных ограничивающих условий. Если рассмотреть её как граф (см. рис. 4.20, *а*), то этот граф является связным и у всех его вершин чётная степень, кроме двух (вершины

3 и 8). Начиная с вершины 3 с помощью того же по сути алгоритма получаем следующий путь:

$$3 - 4 - 7 - 11 - 10 - 9 - 6 - 2 - 3 - 7 - 10 - 6 - 3 - 10.$$

Затем выбираем, к примеру, вершину 2 в качестве общей с оставшейся частью графа и получаем эйлеров цикл для оставшейся части графа:

$$2 - 1 - 4 - 8 - 11 - 12 - 9 - 5 - 2.$$

Соединяя второй цикл с первым, получаем эйлеров цикл для всего графа (рис. 4.20, *б*):

$$3 - 4 - 7 - 11 - 10 - 9 - 6 - 2 - 1 - 4 - 8 - 11 - 12$$
$$- 9 - 5 - 2 - 3 - 7 - 10 - 6 - 3 - 10.$$

в) Третью фигуру обвести нельзя, поскольку у графа есть более двух вершин с нечётной степенью.

Комментарии. Поскольку размер эйлерова цикла, построенного с помощью этого алгоритма, непредсказуемо изменяется, алгоритм относится к категории «уменьшай на различные числа».

Задача обвести фигуру — стандартная для книг с головоломками. Такое применение теоремы Эйлера придумал Питер Г. Тэт (1831–1901), знаменитый шотландский математик и физик [Pet09, p. 232].

29. Ещё раз о магическом квадрате

Решение. Во-первых, нужно найти значение общей суммы для рассматриваемого магического квадрата. Эта сумма, иногда называемая магической суммой, равна сумме всех чисел в строках, поделённой на число строк: $\frac{1+2+\ldots+9}{3} = 15$. Во-вторых, в центральной клетке должна стоять 5. Действительно, если мы обозначим числа в строках 1, 2 и 3 как a, b, c; d, e, f; g, h, i соответственно и сложим все числа, стоящие во второй строке, втором столбце и двух главных диагоналях, мы получим

$$(d+e+f)+(b+e+h)+(a+e+i)+(g+e+c) =$$
$$= 3e+(a+b+c)+(d+e+f)+(g+h+i) = 3e+3\cdot 15 = 4\cdot 15,$$

что даёт $e = 5$. Нам остаётся расставить вокруг него пары (1,9), (2,8), (3,7) и (4,6).

1		
	5	
		9

	1	
	5	
	9	

Рис. 4.21. Два возможных варианта расположения 1 и 9 при построении магического квадрата 3×3

	1	
	5	
	9	

9	5	1

	9	
	5	
	1	

1	5	9

Рис. 4.22. Четыре возможных позиции 1, 5 и 9 в магическом квадрате 3×3

6	1	8
7	5	3
2	9	4

2	7	6
9	5	1
4	3	8

4	9	2
3	5	7
8	1	6

8	3	4
1	5	9
6	7	2

8	1	6
3	5	7
4	9	2

4	3	8
9	5	1
2	7	6

2	9	4
7	5	3
6	1	8

6	7	2
1	5	9
8	3	4

Рис. 4.23. Восемь магических квадратов третьего порядка

С учётом симметричности таблицы есть только два качественно различных варианта разместить 1 и, соответственно, 9: в углах таблицы или не в углах (см. рис. 4.21).

Но первый вариант не позволит нам построить магический квадрат. Если в верхнем правом углу мы разместим число меньше 5, у нас не получится магическая сумма 15 в первой строке. Если мы разместим там число больше 5, то не получится магическая сумма в последнем столбце.

Таким образом, нам следует отказаться от первой частично заполненной таблицы на рис. 4.21 и сосредоточиться на второй. Есть ещё три варианта разместить 1 и 9 в одной строке или в одном столбце с 5. Эти симметричные варианты показаны на рис. 4.22.

Строку или столбец, содержащие 1, надо заполнить числами 6 и 8, что можно сделать двумя способами. После этого числа в оставшихся ячейках определяются однозначно. На рис. 4.23 показаны все восемь магических квадратов третьего порядка. Разумеется, все они симметричны и могут быть получены из любого из них путём поворота и отражения.

Комментарии. Магические квадраты очаровывали людей в течение тысячелетий с тех пор, как они появились в древнем Китае. Было разработано несколько алгоритмов для построения магических квадратов порядка $n>2$, однако не было найдено формулы, дающей число магических квадратов произвольного порядка. Чтобы узнать больше о магических квадратах, можно почитать несколько монографий (например, [Pic02]) или главы во многих книгах по занимательной математике (например [Kra53, гл. 7]), а также многочисленные сайты в Интернете.

30. Ломание палки

Решение. Минимальное число разломов палки длиной в 100 единиц — это семь.

Поскольку можно разломать несколько частей палки одновременно, нам нужно найти алгоритм, который уменьшал бы размер самого длинного куска палки до размера 1. Из этого следует, что на каждой итерации оптимального алгоритма нужно разломать самую длинную часть (и одновременно все другие части, чей размер больше чем 1) пополам (или примерно пополам, насколько это возможно). Если длина l какого-либо кусочка чётная, то нужно разломать на два кусочка длиной $\frac{l}{2}$. Если l — нечётная и больше 1, нужно разломать на части $\left\lceil \frac{l}{2} \right\rceil = \frac{l+1}{2}$ и $\left\lfloor \frac{l}{2} \right\rfloor = \frac{l-1}{2}$ соответственно. Итерации прекращаются, когда длина самого длинного кусочка палки (а значит, и всех остальных) равна 1.

Число разрезов (итераций) такого оптимального алгоритма для палки длиной в n единиц равно $\lceil \log_2 n \rceil$, что является наименьшим значением k таким, что $2^k \geqslant n$. В частности, для $n=100$ получаем $\lceil \log_2 100 \rceil = 7$, поскольку $2^7>100$ и $2^6<100$.

Комментарии. Решение этой головоломки основано на применении стратегии «уменьшай в два раза». Эта стратегия использовалась также в головоломке «Угадай число» из первой части учебного раздела. Двумерная версия задачи — «*Разрезание прямоугольника*» (№ 54).

31. Трюк с тремя стопками карт

Решение. Нужная карта всегда будет точно в середине стопки, которую выбирает загадавший карту при последнем раскладе.

Обозначим карты в трёх стопках после первого расклада символами a1, a2, ..., a9; b1, b2, ..., b9; c1, c2, ..., c9

Стопка 1	Стопка 2	Стопка 3	Стопка 1	Стопка 2	Стопка 3	Стопка 1	Стопка 2	Стопка 3
a1	b1	c1	b1	b2	b3	b1	b4	b7
a2	b2	c2	b4	b5	b6	a1	a4	a7
a3	b3	c3	b7	b8	b9	c1	c4	c7
a4	b4	c4	a1	a2	a3	b3	b6	b9
a5	b5	c5	a4	a5	a6	a3	a6	a9
a6	b6	c6	a7	a8	a9	c3	c6	c9
a7	b7	c7	c1	c2	c3	b2	b5	b8
a8	b8	c8	c4	c5	c6	a2	a5	a8
a9	b9	c9	c7	c8	c9	c2	c5	c8
	(*а*)			(*б*)			(*в*)	

Рис. 4.24. Иллюстрация решения головоломки «*Трюк с тремя стопками карт*»

(рис. 4.24, *а*). Если загаданная карта находится, скажем, в стопке 1, то после второго расклада стопки будут выглядеть, как на рис. 4.24, *б*. Обратите внимание, что все карты, которые были в первой стопке после первого расклада, находятся точно в трёх средних позициях в каждой стопке. Если загаданная карта находится теперь, скажем, в стопке 3 (т. е. одна из карт a3, a6 или a9), то она будет в точности в середине стопки при последнем раскладе (рис. 4.24, *в*). Указание на стопку, в которой находится выбранная карта после последнего расклада, определяет карту однозначно. Легко видеть, что то же самое было бы, если загаданная кара была бы не в стопках 1 и 3 после первого и второго расклада соответственно.

Комментарии. Многие карточные трюки основаны на общих идеях разработки и анализа алгоритмов. Данная головоломка демонстрирует, как простой анализ выходных данных алгоритма решает поставленную задачу. Болл и Коксетер [Bal87, p. 328] говорят, что этот трюк упоминается в классической книге Баше, написанной в XVII веке [Bac12, p. 143]. Головоломка была также включена в книгу «Математические развлечения» Мориса Крайчика [Kra53, p. 317].

32. Турнир на выбывание

Решение. а) Общее число матчей равно $n-1$: после каждого матча остаётся один проигравший, и необходимо в точности $n-1$ проигравших, чтобы в турнире был один победитель.

б) Если $n=2^k$, общее число раундов равно $k=\log_2 n$: каждый раунд уменьшает число оставшихся игроков в два раза, и раунды продолжаются, пока число оставшихся игроков не уменьшится до 1. Если n не является степенью 2, то ответ — наименьшая степень 2, которая больше или равна n, или, используя стандартное обозначение для округления до ближайшего целого числа, $\lceil \log_2 n \rceil$. Например, для $n = 10$ число раундов равно $\lceil \log_2 10 \rceil = 4$.

в) Игроком, занявшим второе место, может быть любой игрок, который проиграл победителю турнира и больше не проиграл никому. Эти игроки могут сыграть в своём собственном турнире на выбывание, который можно организовать следующим образом. На дереве, представляющем все сыгранные матчи, найдём лист, обозначающий победителя турнира, и проследим путь от этого листа к корню, предполагая, что победитель проиграл в первом матче. Для этого потребуется не более чем $\lceil \log_2 n \rceil - 1$ матчей.

Комментарии. Такой турнир можно рассматривать как алгоритм, и два первых вопроса относятся к количеству шагов в алгоритме, понимаемых или как индивидуальные матчи, или раунды. Разумеется, различная интерпретация того, что считать шагами алгоритма, даёт различный подсчёт шагов. Следует отметить, что деревья, изображающие соревнования, нашли интересное применение в информатике (см. [Knu98]).

Мартин Гарднер включил похожую головоломку в книгу «Есть идея!» [Gar78, p. 6]. В головоломке «*Подсчёт „пустых номеров“*» (№ 116) речь идёт о «пустых номерах», которые дают пропуск игрокам в следующий раунд, поскольку для них не нашлось противника.

33. Магия и псевдомагия

Решение. Для вопросов а) и б) ответы соответственно $n = 3$ и $n \geqslant 3$.

а) Поскольку в центральной ячейке магического квадрата размером 3×3 должно стоять число 5 (см. решение головоломки «*Ещё раз о магическом квадрате*» (№ 29)), то решение задачи существует только для $n = 3$ в виде одного из восьми приведённых квадратов.

б) Для $n = 3$ ответ очевиден, поскольку любой магический квадрат 3×3 является также полумагическим.

Если в левом верхнем углу любой таблицы размером $n \times n$ (где $n > 3$) мы заполним квадрат 3×3 числами от 1 до 9 так, чтобы получился магический квадрат, а затем скопируем цифры из первого столбца в четвёртый, то квадрат 3×3, образованный первыми тремя строками и столбцами 2, 3 и 4, будет полумагическим. Таким же образом, если скопировать первую строку в четвёртую, получится полумагический квадрат из строк 2, 3 и 4 и первых трёх столбцов. Это приводит нас к следующему алгоритму для решения рассматриваемой задачи.

Заполним квадрат 3×3 в левом верхнем углу так, чтобы получить некоторый магический квадрат 3×3. Потом запишем в первые три ячейки столбцов 4, 5, ..., n цифры из первых трёх ячеек столбцов 1, 2, ..., $n-3$ соответственно. Затем заполним строки 4, 5, ..., n содержимым строк 1, 2, ..., $n-3$ соответственно. Пример для $n = 5$ изображён на рис. 4.25.

4	9	2	4	9
3	5	7	3	5
8	1	6	8	1
4	9	2	4	9
3	5	7	3	5

Рис. 4.25. Решение головоломки «*Магия и псевдомагия*» для $n = 5$

С другой стороны, на алгоритм можно смотреть как на замощение данной таблицы одним и тем же магическим квадратом 3×3, причём части квадратов, попадающие за границы таблицы (если такие есть), игнорируются.

Комментарии. Идея алгоритма основана на «подходе по возрастающей» (см. первую часть учебного раздела), который мы применяем, начав с самого простого частного случая $n = 3$.

34. Монеты на звезде

Решение. Максимальное число монет, которое можно разместить, это семь.

Без ограничения общности можно вначале поместить монету на точку 6 и переместить её в точку 1, что мы обозначили как $6 \to 1$ (рис. 4.26). Это действие исключит из использования две линии, соединяющие точку 1 с точками 4 и 6. Согласно логике жадной стратегии, мы должны пытаться размещать монеты так, чтобы минимизировать число исключаемых линий и тем самым максимизировать число линий, которые можно будет использовать для перемещения следующих монет. Это означает, что каждую монету, после

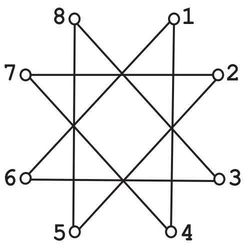


Рис. 4.26. Звезда, на вершинах которой нужно расположить монеты

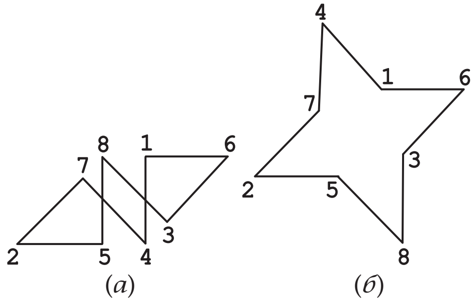


Рис. 4.27. Развёртывание графа, изображённого на рис. 4.26

первой, нужно стараться поместить в свободную точку неиспользуемой линии (передвинув её туда по используемой линии, конечно). Проще всего добиться этого, перемещая каждую монету так, чтобы она оказывалась в точке, из которой начала движение предыдущая монета. Например, семь монет можно разместить с помощью последовательности ходов

$$6 \to 1,\ 3 \to 6,\ 8 \to 3, 5 \to 8,\ 2 \to 5,\ 7 \to 2, 4 \to 7.$$

Очевидно, что мы не можем разместить восемь монет, поскольку после размещения семи монет не останется свободной точки, из которой могла бы двигаться восьмая.

С другой стороны, задачу можно решить путём «развёртывания» графа методом «пуговиц и верёвочек». (Этот метод упомянут в первой части учебного раздела при обсуждении стратегии изменения представления.) Если «взять» вершину 2 графа на рис. 4.26, переместить её в левую часть графа, затем «взять» вершину 6 и переместить её в правую часть графа, получится граф, изображённый на рис. 4.27, *а*. А если затем взять вершины 8 и 4 и перенести их на противоположные стороны графа, получим граф на рис. 4.27, *б*. Из такого представления звезды сразу же следует вышеописанное решение, как и несколько альтернативных решений.

Комментарии. Два изложенных выше решения используют жадную стратегию и один из методов изменения представления графа.

Другие варианты этой задачи, также известной как «Загадка об октограмме» (октограмма — восьмилучевая звезда. — *Прим. перев.*), известны уже на протяжении веков (см. [Sin10, Section 5.R.6]). Её можно найти в современных сборниках головоломок [Dud58, p. 230], [Sch68, p. 15] и [Gar78, p. 38]. Она тесно связана с задачей «*Головоломка Гуарини*», которую мы обсуждали в первой части учебного раздела.

35. Три кувшина

Решение. Задача решается в шесть шагов, показанных на рис. 4.28.

Хотя решение можно получить методом проб и ошибок, есть и другой, систематический способ. Мы можем представить количество воды в кувшинах ёмкостью 3, 5 и 8 литров тройкой неотрицательных целых чисел. Таким образом, начнём с тройки 008. Рассмотрим все допустимые изменения количества воды в кувшинах после переливаний. Для этого воспользуемся понятием «очередь» — одной из главных структур данных в информатике.

Очередь — это последовательность объектов, которая выглядит, как следует из названия, как очередь покупателей к одному кассиру: покупатели обслуживаются в порядке прибытия. Объекты удаляются с одного конца очереди, называемого *началом*, а новые объекты добавляются к другому концу, т. е. к *хвосту* очереди.

В применении к нашей задаче мы организуем очередь с начальной тройки 008 и будем повторять следующие шаги, пока не достигнем в первый раз необходимого состояния — чтобы в тройке была цифра 4.

Пометим компонентами начальной тройки каждую *новую* тройку, которая достижима из начального состояния, добавим новые тройки с метками к очереди и затем уберём начальную тройку. Применение этого алгоритма к данной задаче даёт такую последовательность элементов очереди. Нижние индексы

Номер шага	8-литровый кувшин	5-литровый кувшин	3-литровый кувшин
	8	0	0
1	3	5	0
2	3	2	3
3	6	2	0
4	6	0	2
5	1	5	2
6	1	4	3

Рис. 4.28. Решение головоломки «*Три кувшина*»

показывают метки, когда они появляются в первый раз:

$008|305_{008}, 053_{008}|053, 035_{305}, 350_{305}|035, 350, 323_{053}|350, 323, 332_{035}|$
$323, 332|332, 026_{323}|026, 152_{332}|152, 206_{026}|206, 107_{152}|107, 251_{206}|$
$251, 017_{107}|017, 341_{251}$

Проследив метки от 341 в обратном направлении, получаем последовательность преобразований, которая даёт решение задачи за минимальное число шагов — шесть шагов:

$$008 \to 053 \to 323 \to 026 \to 206 \to 251 \to 341.$$

Комментарии. Решение задачи похоже на так называемый обход «поиск в ширину» [Lev06, гл. 5.2]) графа пространства состояний задачи. Для простоты мы не стали рисовать этот граф в явном виде. Данный алгоритм похож на исчерпывающий перебор.

Некоторые интересные интерпретации и вариации этой старой головоломки можно найти на двух интернет-страничках Математической ассоциации Америки: одна — А. Богомольного [Bog00], где есть ссылка на апплет (небольшая прикладная программа — *прим. перев.*), иллюстрирующий решение головоломки, и другая — Ивара Петерсона [Pet03]. Решить головоломку можно и с помощью неожиданного представления в трёхлинейных координатах, придуманного М. Ч. К. Твиди [Twe39, Chapter 4] (см. также [OBe65]).

36. Ограниченное разнообразие

Решение. Задача имеет решение при чётных n и не имеет решения при нечётных n.

Если n чётно, то можно заполнить верхнюю строку плюсами, строки 2 и 3 — минусами, строки 4 и 5 — снова плюсами, и т. д., пока последний ряд не окажется заполненным. Если действовать таким образом, у каждой клетки будет ровно одна соседняя с противоположным знаком, либо сверху, либо снизу этой клетки. Конечно, можно получить альтернативное решение, поменяв местами плюсы и минусы в предыдущем решении, или же заполняя одинаковыми знаками столбцы, а не строки.

Если в верхнем левом углу написан плюс, то в соседние клетки необходимо поместить один плюс и один минус. Рассмотрим случай, когда плюс записан по соседству в строчку с первым плюсом, а по соседству в столбик с ним записан минус; другой случай симметричен этому. Таким образом, докажем, что если мы запишем плюс в первые две клетки строки 1 и минус в первую клетку строки 2, то остальные клетки строки 1 придётся заполнить плюсами, а остальные клетки строки 2 — минусами. Поскольку у первой клетки второй строки уже есть сосед-плюс сверху, то вторая клетка строки 2 должна содержать минус. Но тогда третья клетка строки 1 должна содержать плюс, поскольку у второй клетки строки 1 уже есть сосед-минус снизу. По такой же логике и все остальные клетки строки 1 должны содержать плюс, а все остальные клетки строки 2 — минус. Но тогда все клетки строки 3 должны содержать минус, поскольку у всех клеток строки 2 уже есть сосед-плюс сверху. Так что строка 3 не может быть последней, за ней должна следовать заполненная плюсами строка 4. Строка 4 может либо быть последней, либо за ней будет следовать ещё строка, заполненная плюсами, и т. д. (Более формально к такому же выводу привела бы математическая индукция.) Это доказывает, что у задачи нет иного решения, кроме вышеизложенного, для чётных n, и нет решения для нечётных n.

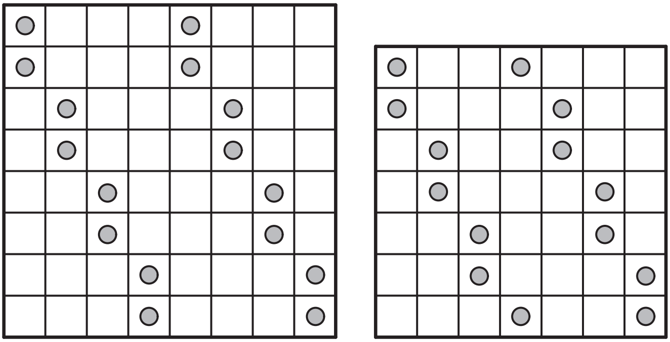
Комментарии. Головоломка является обобщением частного случая для $n = 4$, который предлагается в сборнике А. Спивака [Spi02, задача 67b].

37. Задача о $2n$ шашках

Решение. Поскольку необходимо разместить $2n$ шашек в n строках и n столбцах доски так, чтобы в одной строке или одном столбце было не больше двух шашек, то в каждую строку и в каждый столбец нужно поместить ровно две шашки.

Для чётного $n = 2k$ можно получить решение, одинаковым способом поместив n шашек в первые k столбцов и в последние k столбцов следующим образом. (Мы считаем, что строки и столбцы доски пронумерованы сверху вниз и слева направо соответственно.) Поместим две шашки в первые две строки столбцов 1 и $k+1$, две шашки в строки 3 и 4 столбцов 2 и $k+2$, и т. д., в конце шашки помещаются в строки $n-1$ и n столбцов k и $2k$ (см. рис. 4.29, *а* для случая $n = 8$).

Для нечётного $n = 2k+1$, $k > 0$, можно получить решение, поместив две шашки в строки 1 и 2 столбца 1, две шашки в строки 3 и 4 столбца 2, и т. д., в конце шашки помещаются в строки $n-2$ и $n-1$ столбца k. Затем поместим две шашки в первую и последнюю строку столбца $k+1$. После



(*а*) (*б*)

Рис. 4.29. Решение головоломки «*Задача о 2n шашках*» для (*а*) $n = 8$ и (*б*) $n = 7$

этого нужно поместить k шашек в правую часть доски, симметрично шашкам в левой части относительно центрального квадрата доски: по две шашки — в строки 2 и 3 столбца $k + 2$, в строки 4 и 5 столбца $k + 3$, и т. д., в конце шашки помещаются в строки $n - 1$ и n последнего столбца (см. рис. 4.29, *б* для случая $n = 7$).

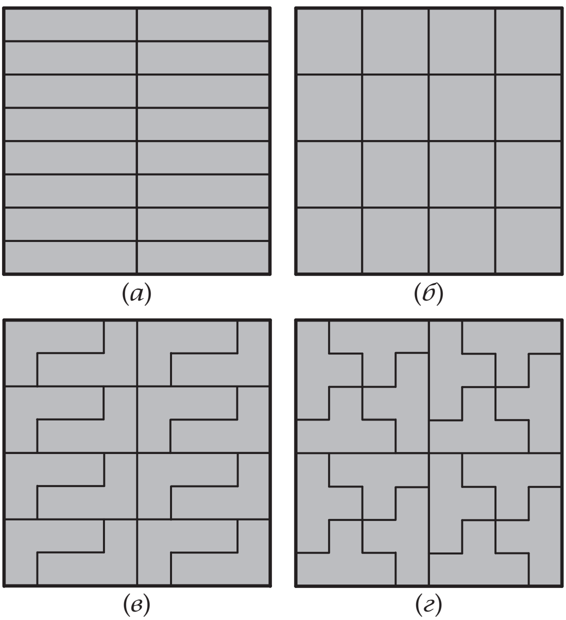
Для $n \geqslant 4$ задачу также можно решить, «смешав» те два решения головоломки «*Снова задача об n ферзях*» (№ 140), в которых ферзи не стоят на одних и тех же клетках доски. Однако не имеет смысла решать головоломку «*Задача о 2n шашках*» таким способом, поскольку она, на самом деле, легче задачи об n ферзях.

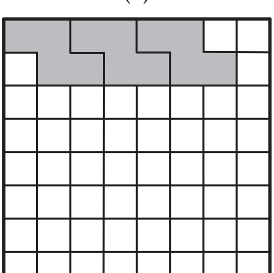
Комментарии. Сэм Лойд [Loy60, задача 48] рассматривает задачу для доски 8×8 с дополнительным условием о том, что две шашки необходимо поместить на два центральных квадрата доски. Генри Дьюдени [Dud58, задача 317] также предлагает более жёсткое условие — о том, что никакие три шашки не должны располагаться на одной прямой (не только на одной строке, столбце или диагонали). В то время как оба автора дают одно и то же решение для доски 8×8, задача в постановке Дьюдени остаётся нерешённой для доски произвольного размера. Для дальнейшего обсуждения задачи см. ч. 5 книги «От мозаик Пенроуза к надёжным шифрам» Мартина Гарднера [Gar97a].

38. Замощение плитками тетрамино

Решение. Способы замощения доски 8×8 прямыми, квадратными, L- и T-образными тетрамино изображены на рис. 4.30, *а–г*. Обратите внимание, что во всех случаях мы повторяем покрытие одной четверти доски для трёх других четвертей.

Покрыть доску 8×8 Z-образными тетрамино невозможно. Действительно, если поместить фигуру такого тетрамино так, чтобы она закрывала угол доски, то необходимо положить ещё две плитки вдоль границы, но после этого будет невозможно закрыть две оставшиеся клетки первого ряда доски (рис. 4.30, *д*).





(*д*)

Рис. 4.30. Покрытие шахматной доски плитками тетрамино. (*а*) Прямые тетрамино. (*б*) Квадратные тетрамино. (*в*) L-образные тетрамино. (*г*) T-образные тетрамино. (*д*) Z-образные тетрамино (нет решения)

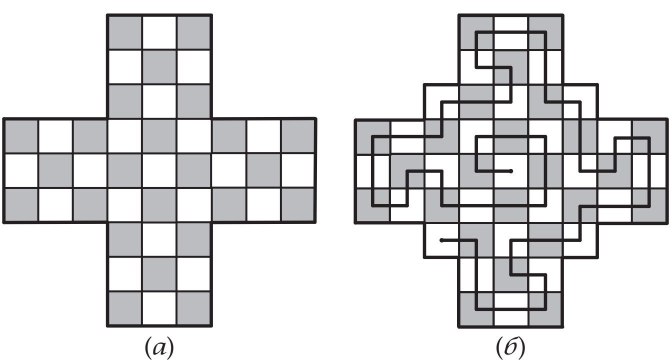
Наконец, невозможно покрыть шахматную доску 8×8 и 15 Т-образными тетрамино и 1 квадратным. Число, например, чёрных полей, которые покрывает одно Т-образное тетрамино — и, тем самым, любое нечётное их число — нечётно, в то время как одно квадратное тетрамино всегда покрывает два чёрных поля. Таким образом, 15 Т-образных тетрамино и 1 квадратное тетрамино покрывают нечётное число чёрных полей, в то время как их общее число на шахматной доске 8×8 чётно.

Комментарии. Получить требуемые замощения прямыми, квадратными, L- и Т-образными тетрамино можно как «в лоб», так и действуя согласно стратегии «разделяй и властвуй». Доказательство невозможности замощения 15 Т-образными тетрамино и 1 квадратным тетрамино, очевидно, основано на идее инварианта (чётность и раскраска).

Эта головоломка — из основополагающей работы Соломона Голомба по замощениям плитками полимино [Gol54].

39. Прогулки по клеточному полю

Решение. Пути, который проходил бы через все клетки доски на рис. 2.11, *а*, не существует. Если раскрасить клетки доски, чередуя цвета, как на шахматной доске (рис. 4.31, *а*), то тёмных клеток будет на три больше, чем светлых. Поскольку цвета клеток на пути должны чередоваться, то путь, проходящий через все клетки доски, изображённой на рис. 4.31, *а*, невозможен.



(*а*) (*б*)

Рис. 4.31. (а) Раскрашивание данной доски. (б) Раскрашивание второй доски. Показан путь, проходящий через все её клетки

Число светлых клеток на доске на рис. 2.11, *б* на 1 больше, чем число тёмных клеток, поэтому путь через все клетки должен начинаться и заканчиваться на светлых клетках. Один такой путь, использующий симметричность доски, показан на рис. 4.31, *б*.

Комментарии. Решение найдено с помощью раскрашивания — одного из наиболее распространённых методов, использующих идею инварианта (см. первую часть учебного раздела).

Обратите внимание, что вопрос задачи заключается в существовании *гамильтонового пути* в графе, чьи вершины представляют собой клетки доски, а рёбра «соединяют» смежные клетки. В отличие от гамильтонового цикла (см. комментарии к задаче «*Игра „Икосиан“*» (№ 27)), гамильтонов путь не должен возвращаться к исходной точке. Однако отсутствие этого требования не делает задачу более лёгкой: эффективный алгоритм для определения существования гамильтонового пути в произвольном графе также неизвестен.

Часть б) основана на задаче 459 коллекции А. Спивака [Spi02].

40. Перестановка четырёх коней

Решение. У головоломки решения нет.

Как мы объясняли в первой части учебного раздела, начальное состояние головоломки удобно представить как граф на рис. 4.32.

Кони могут перемещаться только к соседним вершинам этого графа, сохраняя при этом свой порядок по часовой

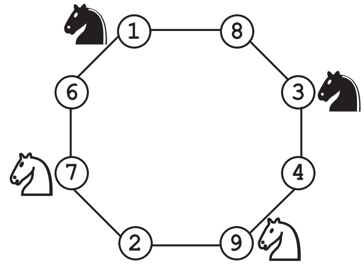


Рис. 4.32. Начальное состояние в головоломке «*Перестановка четырёх коней*», представленное в виде развёрнутого графа

стрелке (или против часовой стрелки): два коня одного цвета, и за ними два коня другого цвета. Поскольку цель головоломки — чтобы кони разных цветов чередовались, решить головоломку нельзя.

Комментарии. В головоломке используется два метода решения алгоритмических задач: изменение представления (граф доски и его развёртывание) и идея инвариантности (чередование коней в порядке по часовой стрелке).

Этот вариант задачи «*Головоломка Гуарини*», о которой говорилось в первой части учебного раздела, взят из книги [Fom96, задача 2, с. 39]. Похожий вариант этой классической головоломки упоминается М. Гарднером в книге «Есть идея!» [Gar78, p. 36].

41. Круг света

Решение. Минимальное количество тумблеров, которые нужно переключить, равно $\frac{n}{3}$, если n кратно 3, и n, если n не делится на 3.

Нетрудно понять, что конечное положение светильников зависит только от того, чётное или нечётное число раз был переключён каждый тумблер, и не зависит от того, в какой последовательности переключались тумблеры. Поэтому всё, что нужно определить, — какие тумблеры нужно переключить один раз, а какие нужно оставить в выключенном положении, чтобы горели все светильники. Чтобы включить светильник, нам нужно или переключить его тумблер, оставив два соседних в начальном положении, или же переключить все три тумблера. Ясно, что по крайней мере один тумблер нужно переключить. Пронумеруем светильники и их тумблеры от 1 до n, например, по часовой стрелке от первого включённого тумблера. Чтобы в итоге горел светильник 1, оба его соседних тумблера (с номерами 2 и n) должны быть или оба включены, или оба выключены. В первом случае тумблеры 3 и $n-1$ тоже должны быть включены. Продолжая далее, мы понимаем, что все тумблеры нужно переключить один раз.

Во втором случае — когда тумблер 1 включён, а тумблеры 2 и n нет — тумблер 3 не должен быть включён, чтобы горел светильник 2; тумблер 4 должен быть включён, чтобы горели светильники 3, а также 4 и 5. Продолжая таким образом, включаем каждый третий тумблер, т. е. $1, 4, \ldots, 3k+1, \ldots, n-2$. Это возможно тогда и только тогда, когда n кратно 3. Для таких n включаем только $\frac{n}{3}$ тумблеров, т. е. меньше, чем все n (тот минимум, который нужен для всех других значений n).

Комментарии. Эту головоломку нам удалось решить «в лоб», но есть её более общая версия, в которой из произвольного начального состояния светильников нужно получить некоторое требуемое состояние, и для решения этой задачи уже нужен более утончённый подход.

Головоломка была опубликована как задача месяца в ноябре 2004 года на *Math Central* — интернет-ресурсе для студентов и преподавателей математики, который поддерживается Университетом Реджайна в Саскачеване, Канада [MathCentral]. На сайте упоминается, что демо-версия головоломки для $n = 7$ была представлена в Музее математики в Гиссене (Германия). Двумерные варианты этой задачи — «Магические квадраты Мерлина» и «Выключить свет» — более известны, чем одномерный.

42. Ещё раз о волке, козе и капусте

Решение. Обозначим волка, козу, капусту и охотника буквами W, G, C и H соответственно.

У задачи есть два симметричных решения:

$$\text{WCWC}\ldots\text{WCHGHG}\ldots\text{HG} \quad \text{и} \quad \text{GHGH}\ldots\text{GHCWCW}\ldots\text{CW}.$$

Основное наблюдение состоит в том, что W может находиться рядом только с C, а G — только с H. При $n = 1$ отсюда немедленно следует, что головоломка имеет два симметричных решения: WCHG и GHCW.

При $n = 2$ решение можно получить, присоединяя WC и GH к началу и концу цепочек для $n = 1$, получая тем самым WCWCHGHG и GHGHCWCW. В общем случае, присоединяя WC и GH $n-1$ раз к началу и концу цепочек для $n = 1$, имеем два решения для любого n, которые и показаны выше.

Другого решения не существует, и это можно объяснить следующим образом. Как и в случае $n = 1$, в каждом решении на двух концах цепочки фишек должны быть фишка W и фишка G. Можем доказать это от противного. Допустим, что есть другое решение. Поскольку на все W и G накладываются симметричные ограничения, без потери общности мы можем предположить, что для такого решения все

n фишек W находятся внутри цепочки. Но тогда в цепочке должны были бы быть $n+1$ фишек C, что невозможно. Таким образом, любое решение должно иметь на концах W и G, а остальные $n-1$ фишек W должны чередоваться с n фишками C в последовательности CWCW...C, а остальные $n-1$ фишек G должны чередоваться с n фишками H, формируя последовательность HGHG...H внутри цепочки решения. В итоге остаётся только один способ разместить CWCW...C и HGHG...H между W и G и только один способ разместить их между G и W, не нарушая при этом ограничения задачи.

Комментарии. Решение головоломки основано на применении метода «уменьшай и властвуй» в обратном порядке: сначала получаем решение для простых частных случаев, а потом распространяем его дальше.

В книге «Математические развлечения» [Kra53, p. 214] М. Крайчик говорит, что авторство головоломки принадлежит Обри, который рассмотрел случай $n = 3$.

43. Расстановка чисел

Решение. Для начала отсортируем список в порядке возрастания чисел. Затем повторим следующую операцию $n-1$ раз: если первый знак неравенства «<», поместим первое (самое маленькое) число в первую коробку. В противном случае поместим туда последнее (самое большое) число. После этого уберём это число из списка и уберём коробочку, в которую поместили это число. Наконец, когда останется единственное число, положим его в оставшуюся коробку.

Комментарии. Этот алгоритм основан на двух стратегиях разработки алгоритмов: «преобразуй и властвуй» (предварительная сортировка) и «уменьшай и властвуй» (уменьшить на единицу). Обратите внимание, что алгоритм не даёт все возможные решения.

Задача была опубликована на веб-страничке [MathCircle].

44. Легче или тяжелее?

Решение. Головоломку можно решить за два взвешивания.

Для начала отложим одну монетку, если n нечётное, и две монетки, если n чётное. Потом разделим оставшееся чётное количество монет на две одинаковые группы и поместим их на чаши весов. Если их вес одинаков, то все монеты — настоящие, и фальшивая монета находится среди отложенных. Поэтому мы можем взвесить отложенную одну или две монеты, сравнив их вес с таким же количеством настоящих монет. Если отложенные монеты весят меньше, то фальшивая — легче. В противном случае она тяжелее.

Если при первом взвешивании веса групп монет разные, возьмём более лёгкую группу и, если число монет нечётное, добавим к ней одну из ранее отложенных монет (которая является настоящей). Разделим эти монеты на две равные группы и взвесим их. Если они весят одинаково, все эти монеты настоящие и поэтому фальшивая монета тяжелее. В противном случае в них есть фальшивая монета и она легче.

Поскольку очевидно, что головоломку нельзя решить за одно взвешивание, этот алгоритм даёт решение за минимально возможное количество взвешиваний.

Комментарии. Головоломка представляет собой очень редкий пример задачи, которую можно решить за одинаковое число основных шагов (а именно за два взвешивания) независимо от размера задачи (в данном случае от количества монет). Другой пример такой головоломки в нашей книге — «*Столбик фальшивых монет*» (№ 11). Варианты этой головоломки включены в книгу Дика Гесса [Hes09, задача 72] и российский сборник задач для школьников [Bos07, с. 41, задача 4].

45. Самый короткий путь коня

Решение. Минимальное количество ходов — 66.

Хотя конь не может пройти к своей цели вдоль прямой линии, он может оставаться на главной диагонали после каждой пары ходов. Таким образом, если начальное и конечное поля — это соответственно (1, 1) и (100, 100), то задача решается последовательностью 66 ходов:

$$(1,1) - (3,2) - (4,4) - \cdots - (97,97) - (99,98) - (100,100)$$

(Число k пар ходов можно получить из уравнения $1 + 3k = 100$.)

С учётом особенности хода коня удобно измерять расстояние между двумя полями доски с помощью так называемого манхэттенского расстояния. Оно рассчитывается как количество горизонтальных рядов плюс количество вертикальных рядов между рассматриваемыми полями. В данной задаче манхэттенское расстояние между начальным и конечным полями равно $(100-1)+(100-1)=198$. Один ход коня может уменьшить это расстояние не более чем на 3, поэтому потребуется по меньшей мере 66 ходов, чтобы конь достиг цели. Это доказывает, что приведённая последовательность ходов действительно является оптимальной.

Комментарии. Можно считать, что решение этой головоломки использует жадную стратегию (см. первую часть учебного раздела), поскольку на каждом шаге алгоритм максимально уменьшает манхэттенское расстояние до цели. Это решение, разумеется, не является единственным. Головоломка «*Куда доскачет конь*» (№ 100) является более общим случаем задачи для доски произвольного размера $n \times n$.

46. Фишки трёх цветов

Решение. При $n = 1$ задача уже решена: все три фишки в единственном столбце — трёх разных цветов.

Для $n > 1$ мы покажем, что фишки можно переставить таким образом, чтобы три фишки в первом столбце были трёх разных цветов, и будем повторять такую перестановку для досок всё меньшего размера, пока задача не будет решена.

Рассмотрим фишки в первом столбце. Есть три варианта: (i) все три фишки разного цвета, (ii) в точности две из них одного цвета и (iii) все одного цвета. В случае (i), очевидно, ничего делать с фишками в первом столбце не нужно.

Рассмотрим случай (ii): без ограничения общности предположим, что две фишки одного цвета — это, скажем, красные фишки и что они находятся в первых двух строках, а фишка в первом столбце и третьей строке, например, белая (см. схему ниже). Поскольку на доске есть n синих фишек и не более чем $n-1$ из них могут находиться в третьей строке, по крайней мере одна из них должна быть в первых двух строках. Так как наш алгоритм может сканировать первые две строки, начиная со столбца 2, пока не встретится синяя фишка, то после её обнаружения синюю и соответствующую красную фишку в первом столбце нужно поменять местами.

Наконец рассмотрим случай (iii). Без ограничения общности предположим, что три фишки в первом столбце — красные. Поскольку в каждой из трёх строк должна быть по крайней мере одна не красная фишка, мы можем просканировать, например, строку 3, чтобы найти эту фишку и поменять её местами с фишкой в первом столбце, тем самым получив случай (ii).

<table>
<tr><td>К</td><td colspan="9" rowspan="2">Здесь должна быть хотя бы одна синяя фишка</td></tr>
<tr><td>К</td></tr>
<tr><td>Б</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Случай (ii)

<table>
<tr><td>К</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>К</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>К</td><td colspan="13">Здесь должна быть синяя или белая фишка</td></tr>
</table>

Случай (iii)

Комментарии. Алгоритм решения задачи использует стратегию «уменьшить на единицу» и стратегию преобразования, о которых говорилось в первой части учебного раздела.

Головоломка взята из сборника А. Спивака [Spi02, задача 670].

47. Планировка выставки

Решение. а) Поскольку посетитель выставки должен пройти через каждый зал в точности по одному разу, он должен входить и выходить из каждого зала через разные двери. Это означает, что минимум 17 дверей должны быть открыты, включая одну дверь для входа на выставку и одну для выхода.

б) Если раскрасить залы как клетки шахматной доски размером 4×4 (рис. 4.33), то становится очевидно, что любой маршрут по выставке должен проходить через клетки чередующихся цветов. Нужно посетить в общем 16 залов, значит, первая и последняя клетка должны быть окрашены в разные цвета. Возможные варианты наружных пар дверей, которые нужно оставить открытыми, — это (A1, B1), (A1, B3), (A2, B2), (A2, B4), а также симметрично (A4, B4), (A4, B2), (A3, B3), (A3, B1). Рисунок 4.34 показывает один маршрут из нескольких возможных для каждой из первых четырёх пар дверей. Разумеется, нужно открыть все двери на пересечении маршрута с границами клеток на плане.

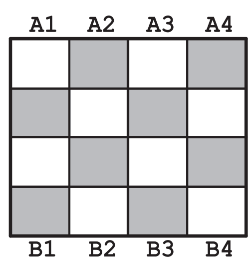


Рис. 4.33. План шестнадцати залов, раскрашенный как шахматная доска

Комментарии. Головоломку можно рассматривать как задачу о гамильтоновых путях в графе, представляющем собой

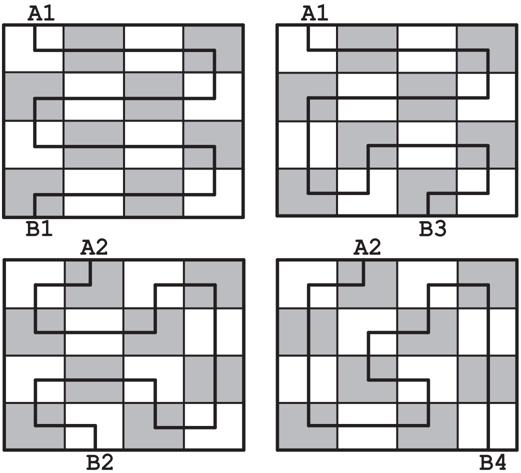


Рис. 4.34. Четыре маршрута через все залы. Обозначены наружные двери для входа и выхода

шахматную доску 4×4, но гораздо проще решить головоломку с помощью стандартного раскрашивания клеток доски.

48. Макнаггет-числа

Решение. а) Очевидно, что следующие шесть чисел не являются макнаггет-числами: 1, 2, 3, 5, 7 и 11.

б) Для любого n, отличного от шести чисел, перечисленных в части (а), n макнаггетсов можно разложить по коробочкам, содержащим 4, 6, 9 и 20 штук с помощью следующего алгоритма. Если $n \leqslant 15$, воспользуемся разложениями:

$$4 = 1 \cdot 4, \quad 6 = 1 \cdot 6, \quad 8 = 2 \cdot 4, \quad 9 = 1 \cdot 9, \quad 10 = 1 \cdot 4 + 1 \cdot 6,$$

$$12 = 3 \cdot 4 (\text{или } 2 \cdot 6), \quad 13 = 1 \cdot 4 + 1 \cdot 9, \quad 14 = 2 \cdot 4 + 1 \cdot 6, \quad 15 = 1 \cdot 6 + 1 \cdot 9.$$

Если $n > 15$, решим задачу этим же методом (т. е. рекурсивно) для $n - 4$ и добавим одну коробочку с 4 наггетсами.

Есть альтернативный, нерекурсивный способ: найдём частное k и остаток r от деления $n - 12$ на 4 (т. е. $n - 12 = 4k + r$, где $k \geqslant 0$ и $0 \leqslant r \leqslant 3$). Это даёт представление n в виде $4k + (12 + r)$. Используя одно из четырёх возможных сочетаний для $12 + r$ и k коробочек по 4 макнаггетса, получаем в общем случае разложение для n макнаггетсов.

Комментарии. Этот алгоритм основан на стратегии «уменьшай (на 4) и властвуй». Коробочки по 20 наггетсов алгоритм не использует, поскольку 20 кратно 4. Очевидно, что в решении, полученном с помощью этого алгоритма, пять коробочек по 4 наггетса можно заменить одной коробочкой с 20 наггетсами.

Отметим, что задача нахождения наименьшего положительного целого числа, которое нельзя представить линейной комбинацией заданного набора натуральных чисел с наибольшим общим делителем 1, известна как *задача Фробениуса о монетах* (см, например, [Mic09, Section 6.7]).

49. Миссионеры и каннибалы

Решение. Задачу можно решить с помощью графа пространства состояний (рис. 4.35).

Четыре варианта маршрута из начальной вершины к целевой, каждый длиной в 11 рёбер, дают решение с минимальным количеством переправ через реку (буква m обозначает миссионера, буква c — каннибала):

2c → c → 2c → c → 2m → mc → 2m → c → 2c → c → 2c
2c → c → 2c → c → 2m → mc → 2m → c → 2c → m → mc
mc → m → 2c → c → 2m → mc → 2m → c → 2c → c → 2c
mc → m → 2c → c → 2m → mc → 2m → c → 2c → m → mc

Комментарии. Для такого решения является достаточным, чтобы грести умели один из миссионеров и два каннибала.

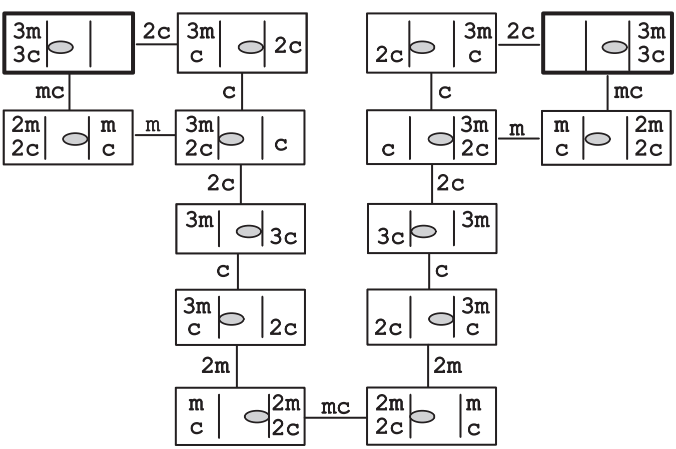


Рис. 4.35. Граф пространства состояний для головоломки «*Миссионеры и каннибалы*». Вершины графа обозначены прямоугольниками, две вертикальные линии обозначают реку, серый овал — расположение лодки. Начальная и целевая вершины выделены жирным. Буквы на рёбрах показывают, кто переправляется через реку (m — миссионер, c — каннибал)

Решение основано на создании графа пространства состояний. Это является стандартным методом решения подобных задач. Другое графическое решение упомянуто в книге [Pet09, p. 253].

Головоломка придумана в XIX веке и является вариантом одной из трёх задач о пересечении реки из средневекового сборника занимательных задач, приписываемого Алкуину Йоркскому (ок. 735–804). Она связана с *задачей о трёх ревнивых мужьях*, версия которой для двух супружеских пар решена в первой части учебного раздела. Другие варианты можно найти в аннотированной библиографии Дэвида Сингмастера [Sin10, Section 5.B].

50. Последний шар

Решение. Последний шар будет чёрным для варианта головоломки а) и белым для варианта б).

а) Нетрудно заметить, что неважно, какого цвета будут два шара, вынутых из мешка; общее количество шаров уменьшится на 1, количество чёрных шаров изменит свою чётность, а количество белых сохранит чётность. Таким образом, если изначально было 20 чёрных и 16 белых шаров, единственный оставшийся шар не может быть белым, поскольку 1 — нечётное число, а 16 — чётное.

б) Если вначале 20 чёрных шаров и 15 белых, последний шар должен быть белым, так как начальное число белых шаров нечётное, а чётность количества белых шаров остаётся неизменной.

Комментарии. Очевидно, что решение основано на инварианте — чётности. Обратите внимание, что в большинстве головоломок инвариант показывает, что головоломка не имеет решения, но в нашем случае это не так. Другую версию головоломки можно найти на [techInt] — одном из сайтов, посвящённых вопросам, задаваемым на собеседовании при приёме на работу.

ГОЛОВОЛОМКИ СРЕДНЕЙ СЛОЖНОСТИ

51. Недостающее число

Решение. Поскольку сумма всех последовательных целых чисел от 1 до 100 равна $S = 1 + 2 + \cdots + 100 = \frac{100 \cdot 101}{2} = 5050$ (см. первую часть учебного раздела, где выводится эта формула), недостающее число m можно найти, вычтя сумму названных чисел $J = 1 + 2 + \cdots + (m-1) + (m+1) + \cdots + 100$ из суммы S: $m = S - J$. То есть Джилл может просуммировать числа, которые называет Джек, а затем подсчитать разность. Например, если недостающее число m равно 10, то $J = 5040$ и m получается как разность $5050 - 5040$.

Суммирование в уме трёхзначных и четырёхзначных чисел — это нелёгкая задача, поэтому алгоритм можно упростить, учитывая только последние два знака всех сумм. Это следует из того наблюдения, что приведённые ниже в таблице все 100 возможных значений J находятся в диапазоне от 4950 до 5049 включительно и могут быть однозначно представлены своими двумя последними знаками. Недостающее число m легко получить с помощью формулы

$$m = \begin{cases} 50 - j, & \text{если} \quad 0 \leqslant j \leqslant 49, \\ 150 - j, & \text{если} \ 50 \leqslant j \leqslant 99, \end{cases}$$

где j — целое число от 0 до 99, образованное двумя последними знаками J. Итак, получаем:

m	1	2	…	49	50	51	…	99	100
J	5049	5048	…	5001	5000	4999	…	4951	4950
j	49	48	…	1	0	99	…	51	50
$50 - j$	1	2	…	49	50				
$150 - j$						51	…	99	100

Более формально эти формулы можно получить из следующего свойства деления по модулю:

$$(S - J) \bmod 100 = (S \bmod 100 - J \bmod 100) \bmod 100,$$

где $S \bmod 100 = 50$ и $j = J \bmod 100$ — это остатки от деления S и J на 100, которые являются числами, образованными двумя последними десятичными знаками S и J соответственно. Далее для значений недостающих m

в диапазоне от 1 до 99 получаем формулу

$$m = m \bmod 100 = (S - J) \bmod 100 = (50 - j) \bmod 100 =$$
$$= \begin{cases} 50 - j, & \text{если } \ 0 \leqslant j \leqslant 49, \\ 150 - j, & \text{если } \ 51 \leqslant j \leqslant 99, \end{cases}$$

и если $m = 100$, мы тоже можем использовать вторую из этих формул, поскольку в этом случае $J = 5050 - 100 = 4950$, $j = 50$, и $m = 150 - j = 100$.

Комментарии. Решение головоломки использует стратегию «преобразуй и властвуй»: мы представляем недостающее число как разность между S и J, а затем представляем эту разность двумя последними цифрами.

Задача о нахождении недостающего числа в последовательности первых натуральных чисел хорошо известна. Данная версия была предложена слушателям Национального общественного радио США в ток-шоу *Car Talk* 6 декабря 2004 г. [CarTalk].

52. Подсчёт треугольников

Решение. После n-й итерации будет $\frac{3}{2}(n-1)n + 1$ маленьких треугольников.

Таблица показывает количество маленьких треугольников $T(n)$ после n итераций алгоритма для первых нескольких значений n:

n	$T(n)$
1	1
2	$1 + 3 = 4$
3	$4 + 6 = 10$
4	$10 + 9 = 19$

Нетрудно заметить (и доказать методом индукции), что количество новых маленьких треугольников, добавляемых на n-й итерации, равно $3(n - 1)$ для $n > 1$. Значит, общее количество маленьких треугольников после n-й итерации можно подсчитать как

$$1 + 3 \cdot 1 + 3 \cdot 2 + \cdots + 3(n-1) =$$
$$= 1 + 3(1 + 2 + \cdots + (n-1)) = \frac{3}{2}(n-1)n + 1.$$

Комментарии. Решение головоломки требует «лобового» применения алгоритмического анализа. Похожий пример описывался во второй части учебного раздела.

Головоломка взята из книги «Математические головоломки» А. Гардинера [Gar99, p. 88, Problem 1].

53. Определение фальшивой монеты с помощью электронных весов

Решение. Минимальное количество взвешиваний для того, чтобы гарантировано найти фальшивую монету, равно $\lceil \log_2 n \rceil$.

Рассмотрим группу S из $m \geqslant 1$ монет, произвольно выбранных из данных n монет. Если вес W группы монет S равен gm, то все монеты настоящие. В противном случае одна из монет в S фальшивая. В первом случае продолжим искать фальшивую монету не в S, а во втором — в S. Если S содержит половину (или около половины) всех монет, после первого взвешивания мы продолжим поиски в вдвое меньшей группе. Нам нужно повторять эту операцию (делить пополам монеты и взвешивать) $\lceil \log_2 n \rceil$ раз, пока размер изначальной группы n не будет гарантировано уменьшен до 1. (Формально число взвешиваний в худшем случае определяется рекуррентным соотношением $W(n) = W\left(\left\lceil \frac{n}{2} \right\rceil\right) + 1$ при $n > 1$, $W(1) = 0$, решением которого является $W(n) = \lceil \log_2 n \rceil$.)

Комментарии. Алгоритм основан на стратегии «уменьшай в два раза». Он практически идентичен алгоритму для игры «*Угадай число* (*двадцать вопросов*)», описанной в первой части учебного раздела.

Задача определения фальшивой монеты — один из наиболее популярных типов головоломок в занимательной математике. Версия с электронными весами встречается реже, чем версия с чашечными весами. Всестороннее обсуждение этой задачи можно найти в статье [Chr84].

54. Разрезание прямоугольника

Решение. Поскольку после каждого вертикального (горизонтального) разреза прямоугольника размером $h \times w$ получается прямоугольник с шириной (высотой) не меньше чем $\left\lceil \frac{w}{2} \right\rceil \left(\left\lceil \frac{h}{2} \right\rceil\right)$, необходимо $\lceil \log_2 n \rceil + \lceil \log_2 m \rceil$ разрезов, чтобы разрезать прямоугольник размером $m \times n$ на mn квадратов размером 1×1. Это количество разрезов является также и достаточным, как показывает следующий алгоритм.

Сначала разрежем $\lceil \log_2 n \rceil$ раз по вертикали вдоль линий все прямоугольники шириной больше 1 (как можно ближе к серединам прямоугольников). Затем разрежем $\lceil \log_2 m \rceil$ раз по горизонтали вдоль линий все прямоугольники высотой больше 1 (как можно ближе к их серединам).

Комментарии. Эта головоломка, так же как её одномерный вариант «*Ломание палки*» (№ 30), использует оптимальность метода «уменьшай наполовину». Обратите внимание, что версия головоломки, в которой не разрешается складывать листы вместе и разрезать более одного прямоугольника за раз, использует совсем другую идею — идею инварианта. Пример такой задачи — «*Разделить плитку шоколада*» — приведён в первой части учебного раздела.

Самое раннее упоминание этой головоломки в библиографии Дэвида Сингмастера [Sin10, разд. 6.AV] относится к 1880 году. В ней нужно разрезать прямоугольник 2×4 на восемь единичных квадратов тремя разрезами. Общий случай головоломки включён в книгу Джеймса Тэнтона [Tan01], где доказательство формулы для минимального числа разрезов проведено по индукции (p. 118).

55. Головоломка «Одометр»

Решение. Ответ на первый и второй вопросы 468 559 и 600 000 соответственно.

Ответ на первый вопрос — это разность между общим количеством показаний прибора и показаниями, в которых нет ни одной цифры 1. Общее количество показаний, очевидно, равно 10^6. Все показания, в которых нет единичек, получаются заполнением шести позиций дисплея одной из девяти цифр; всего таких показаний 9^6. Таким образом, общее количество показаний, в которых есть цифра 1, равно $10^6 - 9^6 = 468\,559$.

Чтобы ответить на второй вопрос, достаточно просто заметить, что каждая из 10 цифр появляется во всех показаниях одинаковое число раз. Поэтому общее число раз, когда появится 1, рассчитывается как одна десятая общего числа всех цифр во всех показаниях: $0{,}1(6 \cdot 10^6) = 600\,000$.

Комментарии. Суть этой головоломки в том, что иногда гораздо проще анализировать алгоритм, рассмотрев специфику задачи, а не применяя общие методы, описанные во второй части учебного раздела. Второй вопрос был задан слушателям Национального общественного радио США в ток-шоу *Car Talk* 27 октября 2008 года [CarTalk].

56. Строй новобранцев

Решение. Швейк понял суть приказа следующим образом: нужно минимизировать выражение

$$\frac{1}{n}\left[(h_2 - h_1) + (h_3 - h_2) + \cdots + (h_n - h_{n-1})\right] = \frac{1}{n}(h_n - h_1), \qquad (1)$$

где n — количество новобранцев, а $h_i, i = 1, 2, \ldots, n,$ — рост новобранца, занимающего в строю i-ю позицию. Поскольку $\frac{h_n - h_1}{n}$ может быть отрицательно, то минимизировать его — значит сделать эту величину отрицательной с наибольшим абсолютным значением. Этого можно достичь, если h_1 и h_n — наибольший и наименьший рост соответственно, а рост остальных людей в выражении (1) не учитывается.

Приказ, конечно же, был другим: минимизировать среднее *абсолютное значение* разности в росте стоящих рядом людей, т. е.

$$\frac{1}{n}(|h_2 - h_1| + |h_3 - h_2| + \cdots + |h_n - h_{n-1}|). \qquad (2)$$

Так как n фиксировано, то множитель $\frac{1}{n}$ можно не учитывать. Сумма (2) минимальна, когда значения роста отсортированы по возрастанию или убыванию, чтобы получить разность $h_{\max} - h_{\min}$ между максимальным и минимальным ростом.

Для любого другого порядка построения эту сумму можно интерпретировать как сумму длин сегментов, покрывающих интервалы с концами на $h_{\min}$ и $h_{\max}$ с некоторым перекрытием, что делает её больше, чем $h_{\max} - h_{\min}$. Нетрудно доказать это более формально с помощью математической индукции.

Комментарии. Сумма $((h_2 - h_1) + (h_3 - h_2) + \cdots + (h_n - h_{n-1}))$ в левой части соотношения (1) называется *телескопическим рядом* из-за физической аналогии с выдвижением телескопа. Иногда это полезное средство для анализа алгоритмов.

В головоломке упоминается герой всемирно известного романа чешского писателя Ярослава Гашека (1883–1923) «Похождения бравого солдата Швейка». Швейк описан как простоватый человек, который, как кажется, готов рьяно исполнять приказы, но в итоге делает так, что всё получается наоборот.

57. Задача Фибоначчи о кроликах

Решение. После 12 месяцев будет 233 пары кроликов.

Пусть $R(n)$ — это количество пар кроликов в конце месяца n. Очевидно, $R(0) = 1$ и $R(1) = 1$. Для каждого $n > 1$ количество пар кроликов, $R(n)$, равно количеству пар в конце месяца $n - 1$, $R(n - 1)$, плюс количество пар кроликов, рождённых в конце месяца n. По условиям задачи количество новорождённых равно $R(n - 2)$ — количеству пар кроликов в конце месяца $n - 2$. Таким образом, получаем рекуррентное соотношение

$$R(n) = R(n-1) + R(n-2) \quad \text{при} \quad n > 1, \quad R(0) = 1, R(1) = 1.$$

В таблице показаны первые 13 членов последовательности, называемые *числами Фибоначчи*, которые определяются этим рекуррентным соотношением:

n	0	1	2	3	4	5	6	7	8	9	10	11	12
$R(n)$	1	1	2	3	5	8	13	21	34	55	89	144	233

Комментарии. $R(n)$ немного отличается от канонической *последовательности Фибоначчи*, которая обычно определяется тем же самым рекуррентным соотношением $F(n) = F(n - 1) + F(n - 2)$, но с другими начальными условиями, а именно $F(0) = 0$ и $F(1) = 1$. Очевидно, $R(n) = F(n + 1)$ при $n \geqslant 0$. Обратите внимание, что мы можем получить R_{12} с помощью одной из двух широко известных формул для чисел Фибоначчи (см., например, [Gra94, разд. 6.6]):

$$R(n) = F(n+1) = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right]$$

и

$$R(n) = F(n+1) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1},$$

округлённое до ближайшего целого.

В этой головоломке задаётся алгоритм и требуется найти его значение на выходе. Это относительно редкий тип алгоритмических головоломок; для большинства из них целью является разработка алгоритма, а не определение выхода данного алгоритма.

Головоломка появилась в «Книге абака» (*Liber Abaci*), написанной итальянским математиком Леонардо Пизанским, известным также как Фибоначчи, в 1202 году. (Другим, более важным вкладом книги в науку стало продвижение индо-арабской системы счисления в Европу.) Последовательность, которая даёт решение задачи, — одна из наиболее интересных и важных последовательностей, когда-либо открытых. Она не только обладает многими удивительными свойствами, но и возникает, часто совсем неожиданно, во многих явлениях природы и областях науки. В наши дни последовательности Фибоначчи и её применению посвящено большое количество книг и веб-сайтов, а также специальный ежеквартальный математический журнал под названием *Fibonacci Quarterly*. В частности, на веб-сайте британского математика Рона Нотта [Knott] есть множество головоломок, связанных с числами Фибоначчи.

58. Сортируем раз, сортируем два...

Решение. Ответ — ноль.

Этот неожиданный результат вытекает из следующего свойства: если есть две последовательности из n элементов $A = a_1, a_2, \ldots, a_n$ и $B = b_1, b_2, \ldots, b_n$ такие, что все элементы A меньше или равны соответствующим элементам B (т. е. $a_j \leqslant b_j$ при $j = 1, 2, \ldots, n$), тогда i-й наименьший элемент в A меньше или равен i-му наименьшему элементу в B для каждого $i = 1, 2, \ldots, n$. Например, если

$$\begin{array}{lllll} A\colon & 3 & 4 & 1 & 6, \\ B\colon & 5 & 9 & 5 & 8, \end{array}$$

то самые маленькие элементы в A и B — соответственно 1 и 5, вторые по меньшинству — 3 и 5, третьи — 4 и 8, четвёртые — 6 и 9. Самый простой способ получить эти значения, конечно же, — отсортировать последовательности:

$$\begin{array}{lllll} A'\colon & 1 & 3 & 4 & 6, \\ B'\colon & 5 & 5 & 8 & 9. \end{array}$$

Чтобы доказать это свойство, рассмотрим $b'_i = b_j$, i-й наименьший элемент в B, который стоит на некоем j-м месте в последовательности B. (Например, если $i = 3$ в нашем примере, $b'_3 = b_4 = 8$, третий по меньшинству элемент в B, который находится на четвёртом месте.) Поскольку b'_i — это i-й по меньшинству элемент в B, то существует $i - 1$ элементов в B, которые меньше или равны b'_i. (В нашем примере это 5 и 5.) Для каждого из этих элементов и самого b'_i, если рассматривать их на изначальных местах в B, существует i элементов на тех же местах в A, которые меньше или равны этим элементам в B. (В нашем примере это 3, 1 и 6.)

Таким образом, b'_i больше или равно (по меньшей мере) i элементам в A. Отсюда получается, что $a'_i \leqslant b'_i$, где a'_i — это i-й наименьший элемент в A: в противном случае a'_i имел бы (по крайней мере) i элементов в A, которые были бы меньше a'_i.

Обозначим теперь через A и B любые два вертикальных ряда карт k и l ($k < l$) после того, как каждый горизонтальный ряд был отсортирован по старшинству карт в первый раз. После сортировки вертикальных рядов массива i-й горизонтальный ряд будет содержать i-ю по меньшинству карту вертикального ряда для любого $i = 1, 2, \dots, n$. В соответствии с этим свойством карта в вертикальном ряду k будет меньше или равна, чем карта в вертикальном ряду l. Поскольку вертикальные ряды k и l были выбраны произвольно, это означает, что горизонтальный ряд i будет отсортирован.

Комментарии. Разумеется, это свойство сохраняется для любых дважды отсортированных двумерных массивов. Дональд Кнут включил эту задачу в третий том книги «Искусство программирования для ЭВМ» [Knu98, p. 238, Problem 27]; он отметил, что задача появилась в книге Германа Бёрнера 1955 года (p. 669). Задача также была включена Питером Уинклером во вторую книгу математических головоломок [Win07, p. 21]; описывая решение, он отметил, что «это одна из тех вещей, которые кажутся то таинственными, то очевидными всякий раз, когда о них думаешь» (p. 24).

59. Шапки двух цветов

Решение. Предположим, что только одна шапка чёрная. Тогда заключённый, на котором она надета, будет видеть на других заключённых только белые шапки. Так как он знает, что должна быть по крайней мере одна чёрная шапка, значит, эта шапка на нём. Каждый из других заключённых видит одну чёрную шапку и не может однозначно понять, какого цвета его шапка. Поэтому заключённый, который видит только белые шапки, — единственный, кто делает шаг вперёд при первом построении. И все освобождены.

Предположим, что есть две чёрных шапки. В этом случае при первом построении ни один заключённый не делает шаг вперёд, поскольку никто не уверен в цвете своей шапки. Но во время второго построения оба заключённых, которые видят одну чёрную шапку, делают шаг вперёд. В самом деле, если ни один заключённый не сделал шаг вперёд во время первого построения, они знают, что есть по крайней мере две чёрных шапки. Если какой-нибудь заключённый видит только одну чёрную шапку, он понимает, что вторая шапка на нём. Все заключённые, которые видят две чёрных шапки, не могут быть уверены в цвете своей шапки и поэтому они остаются в ряду.

В общем случае, если есть k чёрных шапок, $1 \leqslant k \leqslant 11$, ни один из заключённых не выступит во время первых $k-1$ построений. Но при k-м построении каждый заключённый, который видит $k-1$ чёрных шапок, может сделать шаг вперёд по той же причине: он знает, что есть по крайней мере k чёрных шапок, поскольку на предыдущем построении испытание не окончилось. Если он видит $k-1$ чёрных шапок и их в точности k, то одна из них на нём. В то же время все $n-k$ заключённых в белых шапках должны и будут (с учётом их сообразительности) оставаться в строю, поскольку они не могут определить цвет своей шапки. Умные должны получить в конце концов свободу!

Комментарии. Решение построено на стратегии «уменьшай и властвуй», которая применена в обратном порядке по числу чёрных шапок.

Было опубликовано несколько версий этой задачи, как в печати, так и в электронном виде. Приведённая версия взята, в несколько изменённой формулировке, из книги У. Паундстоуна [Pou03, p. 85]. Самый ранний вариант о трёх людях с пятнами на лбу появился около 1935 года и приписывается известному принстонскому логику Алонзо Чёрчу. В нашей книге есть похожая, но более трудная головоломка «*Шапки с номерами*» (№ 147).

60. Переделка треугольника из монеток в квадрат

Решение. Минимальное число монеток, которые нужно передвинуть, равно $\left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil$. У головоломки есть два решения для любого чётного $n > 2$, одно решение для любого нечётного $n > 1$ и три решения для $n = 2$.

Поскольку общее число монеток равно квадрату числа n, $S_n = \sum_{j=1}^{n} \left(2j - 1\right) = n^2$, в каждом из n горизонтальных рядов, составляющих искомый квадрат, должно быть n монет. Естественно создать такой квадрат, выравнивая ряды монет в треугольнике, т. е. передвигая лишние монеты из $\left\lfloor \frac{n}{2} \right\rfloor$

Рис. 4.36. Два симметричных решения для S_4 монет. Добавленные и убранные монеты показаны знаками + и – соответственно. Монеты, оставшиеся на своём месте, обозначены чёрными точками

рядов, в которых больше n монет, в $\left\lfloor \frac{n}{2} \right\rfloor$ рядов, в которых меньше n монет. Это можно сделать, передвинув $n-1$ монет из самого длинного ряда (т. е. гипотенузы), в котором $2n-1$ монет, в ряд с одной монетой. Затем передвинуть $n-3$ монет из следующего ряда, в котором $2n-3$ монет, в ряд с тремя монетами, и т. д., как показано на рис. 4.36 и 4.37 для случаев чётного и нечётного n соответственно. (На этих рисунках заданный треугольник нарисован прямым углом вверх, гипотенуза горизонтальна, а ряды монеток параллельны ей.) Очевидно, что любой алгоритм, который на каждом шаге берёт монетку из ряда, который нужно укоротить, и передвигает её в ряд, который нужно удлинить, решает задачу за минимальное число ходов.

Общее число монет $M(n)$, которые нужно переставить для такой трансформации, можно определить по формуле

$$M(n) = \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor}(n-(2j-1)) = \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor} n - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor}(2j-1) =$$

$$= n\left\lfloor \frac{n}{2} \right\rfloor - \left\lfloor \frac{n}{2} \right\rfloor^2 = \left\lfloor \frac{n}{2} \right\rfloor \left(n - \left\lfloor \frac{n}{2} \right\rfloor\right) = \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil.$$

Единственный возможный вариант, который можно было бы сделать за меньшее число перестановок, — это квадрат, составленный из рядов с нечётными номерами, которые параллельны одному из катетов треугольника. Но для этого потребовалось бы передвинуть все монеты в рядах с чётными номерами, общее число которых равно

$$\bar{M}(n) = \sum_{j=1}^{n-1} j = \frac{(n-1)n}{2} > \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{n}{2} \right\rceil \quad \text{при } n > 2.$$

При $n = 2$, однако, $\bar{M}(2) = M(2) = 1$ и у задачи есть третье, совершенно не похожее на остальные, решение, которое показано на рис. 4.38.

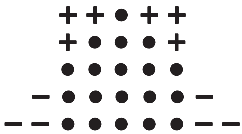


Рис. 4.37. Решение для S_5 монет

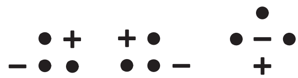


Рис. 4.38. Три решения для S_2 монет

Комментарии. Вариант С. Грабарчука для этой задачи был опубликован на веб-сайте *Puzzles.com* [Graba].

61. Шашки на диагонали

Решение. У головоломки есть решение тогда и только тогда, когда или $n-1$ кратно 4, или n кратно 4. Число перестановок равно $\frac{(n-1)n}{4}$.

Мы можем измерить расстояние между текущим положением шашек и их положением в нижнем ряду как сумму клеток ниже всех шашек. В начальном положении это расстояние равно $(n-1)+(n-2)+\ldots+1 = \frac{(n-1)n}{2}$. В конечном положении оно равно 0. Поскольку каждый шаг уменьшает эту сумму ровно на 2, чётность суммы не изменяется. Поэтому, чтобы у задачи было решение, необходимо, чтобы $\frac{(n-1)n}{2}$ было чётным.

Отметим, что $\frac{(n-1)n}{2}$ чётное тогда и только тогда, когда или $n-1$ кратно 4, или n кратно 4. В самом деле, если $\frac{(n-1)n}{2} = 2k$, то $(n-1)n = 4k$, а поскольку или $n-1$, или n нечётное, то второе из них должно быть кратно 4. Напротив, если или $n-1$, или n кратно 4, то $\frac{(n-1)n}{2}$ является, очевидно, чётным.

Таким образом, чтобы у задачи было решение, необходимо, чтобы или $n = 4k$, или $n = 4k+1$, где $k \geqslant 1$. Это условие является также достаточным, так как задача может быть решена с помощью следующего алгоритма. Передвинем $\left\lceil \frac{n-2}{2} \right\rceil$ пар шашек во всех последовательных столбцах подряд — т. е. в столбцах 1 и 2, 3 и 4 и т. д. — как можно ниже. Затем передвинем $\left\lfloor \frac{n}{4} \right\rfloor$ пар в последовательных нечётных

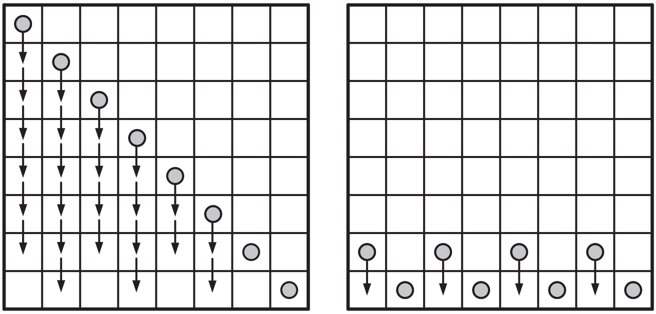


Рис. 4.39. Как передвинуть $n = 8$ шашек с диагонали в нижний ряд (две шашки за раз)

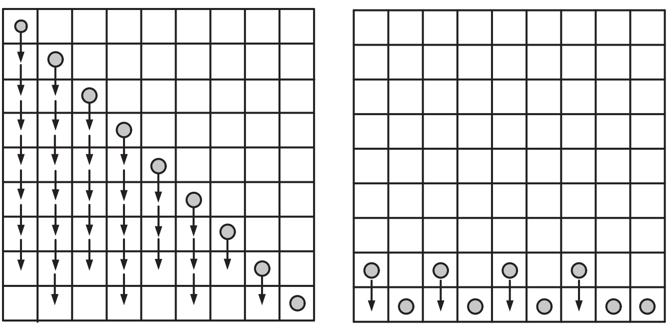


Рис. 4.40. Как передвинуть $n = 9$ шашек с диагонали в нижний ряд (две шашки за раз)

столбцах — т. е. 1 и 3, 5 и 7 и т. д. — на одну клетку. (Заметим, что если n или $n-1$ кратно 4, то существует чётное число нечётных чисел, меньших n или $n-1$ соответственно.) Алгоритм показан на рис. 4.39 и 4.40.

В итоге любой алгоритм решения задачи, в том числе и приведённый выше, уменьшает расстояние между начальной и конечной позициями на 2 на каждой итерации. Таким образом, каждый алгоритм делает одинаковое число шагов: $\frac{(n-1)n}{4}$.

Комментарии. Решение использует формулу суммы первых последовательных целых чисел и чётность как идею инварианта. О них говорилось во второй части учебного раздела.

Головоломка является общим случаем задачи 448 в книге [Spi02], где ставится вопрос о существовании решения при $n = 10$.

62. Робот собирает монетки

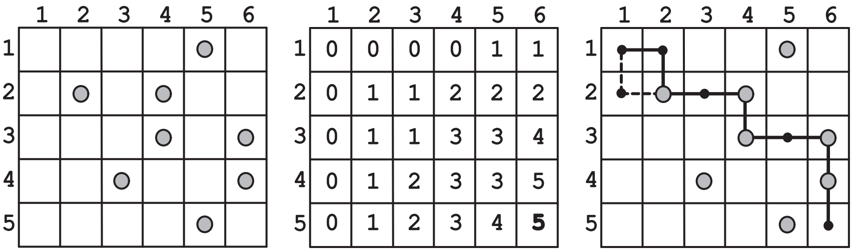
Решение. Как упоминалось в подсказке, для решения задачи можно использовать динамическое программирование. Пусть $C[i,j]$ — наибольшее количество монет, которое робот может собрать и принести на клетку (i,j) в i-й строке и j-м столбце доски. Он может перейти на эту клетку или с соседней сверху клетки $(i-1,j)$, или с соседней слева клетки $(i,j-1)$. Наибольшее количество монет, которое он может принести на эти клетки, — $C[i-1,j]$ и $C[i,j-1]$ соответственно. (Разумеется, у клеток первой строки нет соседних клеток сверху, а у клеток первого столбца нет соседних клеток слева. Для этих клеток мы примем, что у несуществующих соседей $C[i-1,j]$ и $C[i,j-1]$ равно 0.) Таким образом, максимальное количество монет, которое робот может иметь на клетке (i,j), — это максимум этих двух величин плюс, может быть, ещё одна монетка на самой клетке (i,j). Другими словами, для $C[i,j]$ у нас есть такая формула:

$$C[i,j] = \max\{C[i-1,j],\ C[i,j-1]\} + c_{ij} \quad \text{при } 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m, \qquad (1)$$

где $c_{ij} = 1$, если на клетке (i,j) лежит монета, и $c_{ij} = 0$ в противном случае, а $C[0,j] = 0$ при $1 \leqslant j \leqslant m$ и $C[i,0] = 0$ при $1 \leqslant i \leqslant n$.

Используя эти формулы, мы можем заполнить таблицу размером $n \times m$ значениями $C[i,j]$, строка за строкой или столбец за столбцом, как обычно делается в алгоритмах динамического программирования. Это показано на рис. 4.41, *б* для исходного расположения монет на рис. 4.41, *а*.

Чтобы найти путь, на котором можно собрать наибольшее количество монет, нам нужно определить, обусловлен ли максимум в выражении (1) значением на клетке сверху или значением на клетке слева: в первом случае путь должен идти вниз от этой клетки, во втором — слева. При равенстве значений оптимальный путь не является единственным и каждый вариант даёт шаг на некотором оптимальном пути. Например, рис. 4.41, *в* показывает два таких оптимальных пути при расположении монеток, изображённом на рис. 4.41, *а*.



(*а*) (*б*) (*в*)

Рис. 4.41. (*а*) Монеты на доске. (*б*) Результат применения алгоритма динамического программирования. (*в*) Два пути, позволяющие собрать максимально возможное количество монет — 5 монет

Комментарии. Головоломка является упрощённой версией задачи в статье [Gin03]. Она представляет собой хороший пример применения динамического программирования.

63. Плюсы и минусы

Решение. У головоломки есть решение тогда и только тогда, когда или n, или $n+1$ делится на 4.

Задача эквивалентна разделению n целых чисел от 1 до n на два непересекающихся (т. е. без общих элементов) подмножества с одинаковой суммой: подмножество чисел с плюсами перед ними и подмножество чисел с минусами перед ними. Поскольку $S = 1 + 2 + \ldots + n = \frac{n(n+1)}{2}$, сумма чисел в каждом подмножестве должна быть равна в точности половине S. Поэтому число $\frac{n(n+1)}{2}$ должно быть чётным; это является необходимым условием того, чтобы головоломка имела решение. Как показано ниже, для существования решения данное условие является также и достаточным.

Отметим, что $\frac{n(n+1)}{2}$ чётное тогда и только тогда, когда или n кратно 4, или $n+1$ кратно 4. В самом деле, если $\frac{n(n+1)}{2} = 2k$, тогда $n(n+1) = 4k$, а поскольку или n, или $n+1$ чётное, то оно и должно делиться на 4. Верно и обратное, если n или $n+1$ кратно 4, то $\frac{n(n+1)}{2}$, очевидно, чётное.

Если n кратно 4, мы можем, например, разделить последовательность целых чисел от 1 до n на $\frac{n}{4}$ групп, и затем в каждой группе добавить плюсы перед первым и четвёртым числом и минусы перед вторым и третьим числом:

$$(1-2-3+4)+\cdots+((n-3)-(n-2)-(n-1)+n)=0. \qquad (1)$$

Если $n+1$ делится на 4, тогда $n=4k-1=3+4(k-1)$ и мы можем использовать эту же идею, сначала выделив первые три числа, следующим образом:

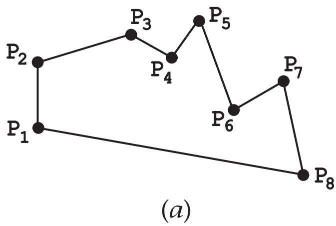
$$(1+2-3)+(4-5-6+7)+\cdots+((n-3)-(n-2)-(n-1)+n)=0. \qquad (2)$$

Таким образом, задачу можно решить с помощью следующего алгоритма. Вычислим $n \bmod 4$ (остаток от деления n на 4). Если остаток равен 0, вставим знаки «+» и «−» согласно формуле (1). Если остаток равен 3, вставим знаки «+» и «−» согласно формуле (2). В противном случае алгоритм выдаёт сообщение «решения нет».

Комментарии. Это — алгоритмическая версия хорошо известной головоломки, которая использует несколько идей, включая широко применяемую формулу суммы последовательных целых чисел от 1 до n и её чётность. Отметим, что для задачи разбиения произвольной последовательности целых чисел на подгруппы с одинаковой суммой эффективный алгоритм неизвестен. Более того, большинство специалистов в области вычислений считают, что эффективного алгоритма не существует, потому что известно, что задача разбиения NP-полна.

64. Восьмиугольники

Решение. Без ограничения общности можно считать, что точки пронумерованы слева направо от 1 до n. Если точки расположены на одном уровне, тогда сначала пронумеруем нижнюю точку. (Формально это означает, что точки упорядочены в соответствии с первой координатой на плоскости в декартовой системе координат.) Рассмотрим первые восемь точек $p_1, \ldots, p_8$ и проведём прямую линию, соединяющую p_1 и p_8. Возможны два случая: или все остальные шесть точек $p_2, \ldots, p_7$ лежат по одну сторону от этой прямой (рис. 4.42, *а*), или по обе стороны (рис. 4.42, *б*). В первом случае восьмиугольник можно нарисовать, соединив по порядку точки $p_1, \ldots, p_8$, причём линия, соединяющая p_1 и p_8, тоже будет одной из сторон восьмиугольника. Если же точки $p_2, \ldots, p_7$ лежат по обе стороны от прямой между p_1 и p_8, то получить восьмиугольник можно следующим образом. Соединим последовательно точки на этой прямой или



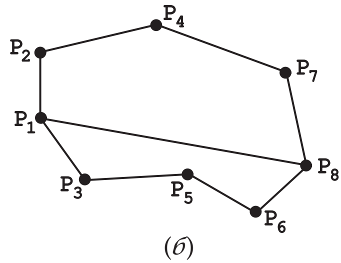


Рис. 4.42. Строим простой восьмиугольник с вершинами в восьми заданных точках

выше неё и получим верхнюю границу восьмиугольника; соединим последовательно точки на этой прямой или ниже неё и получим нижнюю границу (рис. 4.42, *б*).

Отметим, что или все оставшиеся 1992 точки лежат справа от построенного восьмиугольника, или только одна самая левая точка p_9 лежит на той же вертикали, что и p_8, которая является самой правой точкой построенного восьмиугольника. Таким образом, используя тот же метод, мы можем построить восьмиугольник с вершинами в следующих восьми точках $p_9, \ldots, p_{16}$, и он не будет пересекаться с первым восьмиугольником. Повторим этот шаг для каждых последовательных восьми точек, и задача решена.

Комментарии. Для решения задачи используются методы предварительной сортировки, «разделяй и властвуй» и «уменьшай и властвуй». Решение похоже на хорошо известный в информатике «алгоритм быстрой оболочки» (*quickhull*). Он применяется для построения *выпуклых оболочек* заданных наборов точек (см., например, [Lev06, разд. 4.6]).

65. Угадывание кода

Решение. Для определения кода $b_1 b_2 \ldots b_n$ можно использовать такую последовательность битовых строк в n последовательных вопросах:

$$000\ldots0,\ 100\ldots0,\ 110\,0, \ldots,\ 11\ldots10.$$

Первый ответ, a_1, даёт общее количество нулей в коде, который нужно угадать. Пусть a_2 — это ответ на второй вопрос. Поскольку битовые строки в первых двух вопросах отличаются только первыми битами, соответствующее число угаданных битов, a_1 и a_2, отличается на единицу. Поэтому определяем первый бит в коде, b_1, следующим образом: если $a_1 < a_2$, то $b_1 = 1$; если $a_1 > a_2$, то $b_1 = 0$. (Например, для кода 01011 имеем $a_1 = 2$ и $a_2 = 1$; следовательно, $b_1 = 0$, поскольку $a_1 > a_2$.) Повторяя эту процедуру для следующих $n-2$ вопросов, мы сможем определить поочерёдно биты в коде $b_3, \ldots, b_{n-1}$. Последний бит b_n можно найти, используя общее количество нулей в коде, которое мы определили после первого вопроса. Если это количество равно количеству нулей в первых $n-1$ битах (которое нам теперь известно), то $b_n = 1$; в противном случае $b_n = 0$.

На самом деле можно начать с любой n-битовой строки и задать n вопросов, меняя каждый раз один бит.

Комментарии. Головоломка похожа на популярную игру «*Mastermind*» (вариация «Быков и коров»), она взята из книги Денниса Шаши [Sha07]. Приведённое решение основано на стратегии «уменьшай и властвуй». Оно не является оптимальным для любых значений n. Например, любой 5-битовый код можно определить последовательностью 00000, 11100, 01110, 00101 (см. [Sha07, p. 105–106]). Общая формула для минимального числа вопросов при произвольном n неизвестна.

66. Оставшееся число

Решение. Оставшееся число может быть любым положительным нечётным целым числом меньше 50.

Начальная сумма всех чисел на доске равна нечётному числу $1+2+\ldots+50 = 1275$. Каждая операция по замене a и b на $|a-b|$, где без потери общности $a \leqslant b$, уменьшает сумму на $2a$:

$$S_{\text{new}} = S_{\text{old}} - a - b + |a-b| = S_{\text{old}} - b - a + b - a = S_{\text{old}} - 2a.$$

Из этого следует, что новая сумма должна быть нечётной, если начальная была нечётной, и поэтому если в начале было число 1275, в результате повторения данной операции не может остаться чётное число.

Далее, все числа на доске неотрицательные. Они также меньше или равны 50, поскольку $|a-b|$ всегда меньше или равно максимуму из a и b для неотрицательных a и b.

Покажем, что любое нечётное целое число от 1 до 49 включительно можно получить, выполнив операцию головоломки 49 раз. Пусть k — это такое число. Мы можем получить его на первой итерации, вычитая 1 из $k+1$. Затем применим операцию к парам оставшихся последовательных целых чисел

$$(2,3),(4,5),\ldots,(k-1,k),\ (k+2,k+3),\ldots,\ (49,50),$$

что заменит их 24 единицами. Применив к ним операцию головоломки, получим 12 нулей; повторив действие 11 раз, получим один ноль. Наконец, применив действие к оставшимся k и 0, получим k.

Комментарии. Эта головоломка является версией задачи, хорошо известной в математических кругах и основанной на принципе чётности. Головоломка «*Плюсы и минусы*» (№ 63) в нашей книге — это другой вариант на эту же тему.

67. Разливаем пополам

Решение. Если последовательно разделять поровну воду между данным непустым сосудом и каждым из девяти пустых сосудов, то в конечном итоге в нём останется $\frac{a}{2^9}$ литров воды. Это минимальное количество воды, которое может остаться в сосуде. В самом деле, пусть m — минимальное положительное количество воды среди всех сосудов в текущий момент. (Изначально $m = a$, и наша цель — минимизировать его.) Поскольку среднее двух чисел всегда больше или равно меньшему из них, значение m может быть уменьшено путём осреднения, только если у нас есть сосуд, содержащий m литров, и пустой сосуд. После повторения операции осреднения с каждым пустым сосудом пустых сосудов не останется и уменьшение m станет невозможным. Таким образом, минимальное значение m, которое мы можем получить, равно $\frac{a}{2^9} = \frac{a}{512}$ литров.

Комментарии. Головоломка решается применением жадной стратегии. Она основана на идее полуинварианта, о которой говорилось в первой части учебного раздела. Головоломка была включена (в другой формулировке) в статью в журнале «Квант» о полуинвариантах [Kur89, задача 6] после того, как она предлагалась на Ленинградской математической олимпиаде в 1984 году.

68. Сумма цифр

Решение. Общая сумма цифр в целых числах от 1 до 10^n включительно равна $45n \cdot 10^{n-1} + 1$. В частности, для $n = 6$ она равна $27\,000\,001$.

Удобно не рассматривать число 10^n, которое, очевидно, прибавляет 1 к сумме, и вычислить сумму цифр всех целых чисел от 1 до $10^n - 1$. Обозначим эту сумму $S(n)$. Также удобно ввести в запись каждого целого числа меньше $10^n - 1$ необходимое количество нулей в первых разрядах так, чтобы все числа состояли из n цифр.

Простой способ найти $S(n)$ — объединить попарно числа 0 и $10^n - 1$, 1 и $10^n - 2$, 2 и $10^n - 3$, и т. д., поскольку сумма цифр в каждой паре равна $9n$. (Этот трюк использовался во второй части учебного раздела для подсчёта суммы первых натуральных чисел.) Очевидно, что есть $\frac{10^n}{2}$ таких пар, поэтому сумма $S(n)$ равна $9n \cdot \frac{10^n}{2} = 45n \cdot 10^{n-1}$.

Можно также использовать подход, который применялся для решения задачи «*Головоломка „Одометр“*» (№ 55). С учётом нулей в первых разрядах у нас есть 10^n n-разрядных целых чисел, в которых каждая из 10 цифр встречается одинаковое количество раз $n \cdot \frac{10^n}{10} = n \cdot 10^{n-1}$. Таким образом, $S(n)$, общая сумма цифр в них, может быть подсчитана как

$$0 \cdot n \cdot 10^{n-1} + 1 \cdot n \cdot 10^{n-1} + 2 \cdot n \cdot 10^{n-1} + \dots + 9 \cdot n \cdot 10^{n-1} =$$
$$= (1 + 2 + \dots + 9)n \cdot 10^{n-1} = 45n \cdot 10^{n-1}.$$

Третий способ решить задачу — установить рекуррентное соотношение для $S(n)$. Очевидно, что $S(1) = 1 + 2 + \ldots + 9 = 45$. Для каждых возможных самых левых десяти цифр в n-разрядном целом числе общая сумма остальных цифр равна $S(n - 1)$. Сумма самых левых цифр равна $10^{n-1}(0 + 1 + 2 + \ldots + 9) = 45 \cdot 10^{n-1}$. Таким образом, получаем рекуррентное соотношение:

$$S(n) = 10S(n-1) + 45 \cdot 10^{n-1} \quad \text{при} \quad n > 1,\ S(1) = 45.$$

Решая это соотношение методом обратной замены (см. вторую часть учебного раздела), получаем $S(n) = 45n \cdot 10^{n-1}$.

Итак, при $n = 6$ имеем $S(6) = 45 \cdot 6 \cdot 10^{6-1} = 27\,000\,000$ и общая сумма цифр равна $27\,000\,001$.

Комментарии. Объединение чисел в пары с одинаковой суммой цифр можно рассматривать как изменение представления или инвариант, как сделано в книге [Mic08, p. 61–62]. Метод попарного объединения был также использован Б. А. Кордемским [Kor72, p. 202]. Решение с помощью рекурсии основано на стратегии «уменьшай на единицу».

69. Фишки на секторах круга

Решение. У задачи есть решение тогда и только тогда, когда n или нечётно, или кратно 4.

Пронумеруем все секторы от 1 до n, например по часовой стрелке, начиная с любого сектора. Если n нечётное, все фишки можно переместить на средний сектор (т. е. сектор $j = \frac{1+n}{2}$), просто передвигая вместе каждую пару фишек, находящихся на одинаковом расстоянии от среднего сектора — т. е. фишки из секторов 1 и n, 2 и $n-1, \ldots, j-1$ и $j+1$ — до тех пор, пока каждая пара не достигнет его.

Пусть теперь n чётно. Рассмотрим сумму S номеров секторов, на которых в данный момент есть фишки. (Если на одном секторе лежат несколько фишек, то номер сектора включается в сумму несколько раз, по одному разу на каждую фишку.) Легко видеть, что чётность S остаётся неизменной после каждого хода, при котором две фишки перемещаются на соседние секторы. В самом деле, перемещение одной фишки на соседний сектор изменяет чётность суммы, а перемещение двух фишек сохраняет. Если у задачи есть решение, т. е. все фишки могут быть передвинуты на один сектор j ($1 \leqslant j \leqslant n$), то конечное значение $S = nj$ будет чётным, поскольку мы рассматриваем чётный случай n. Следовательно, в начальном положении сумма $1+2+\ldots+n = \frac{n(n+1)}{2}$ тоже должна быть чётной: $\frac{n(n+1)}{2} = 2k$. Но тогда $n(n+1) = 4k$ и, поскольку $n+1$ нечётное, n должно делиться на 4. Это доказывает необходимость данного условия для существования решения при чётном n. Докажем, что оно является и достаточным. Если n кратно 4, мы можем, к примеру, начать передвигать все фишки с нечётными номерами секторов на соседние секторы с чётными номерами, по две за раз в направлении по часовой стрелке. (Существует чётное количество нечётных номеров, меньших чем $n = 4i$.) Затем мы можем передвинуть пары фишек из сектора 2 в сектор n, сделать то же самое для пары из сектора 4, и т. д., пока все фишки не будут собраны в последнем секторе.

Комментарии. Головоломка использует идею чётности и является общим случаем задачи 2 в [Fom96, p. 124].

70. Прыжки в пары — 1

Решение. Очевидно, что для нечётного количества монет у задачи нет решения, поскольку в конце все монеты должны быть собраны по парам, что даёт чётное общее количество монет. Рассматривая все возможные ходы, нетрудно заметить, что задача не имеет решения при $n = 2$, 4 и 6. При $n = 8$ у задачи есть несколько решений. Одно из них можно найти методом поиска с возвратом: 4 на 7, затем 6 на 2, потом 1 на 3 и 5 на 8. Любой частный случай для чётного числа монет более 8 (т. е. $n = 8 + 2k$, где $k > 0$) может быть упрощён до случая числа монет, уменьшенного на два (т. е. $n = 8 + 2(k - 1)$). При этом монета в позиции $8 + 2k - 3$ перемещается на последнюю монету в позиции $8 + 2k$, что и уменьшает размер задачи на 2. Повторив это действие k раз, мы уменьшим задачу размером $n = 8 + 2k$ монет до задачи с 8 монетами, решение которой приведено выше.

Этот алгоритм, очевидно, делает минимальное число шагов, поскольку формирует пары монет на каждом шаге.

Комментарии. Алгоритм использует метод поиска с возвратом и стратегию «уменьшай и властвуй». Самое первое упоминание этой головоломки встречается в Японии в 1727 году, согласно библиографии Дэвида Сингмастера [Sin10, Section 5.R.7]. Головоломка, в числе прочих, обсуждалась в книгах Болла и Коксетера [Bal87, p. 122] и Мартина Гарднера [Gar89]. Мартин Гарднер назвал её «самой древней и лучшей головоломкой про монеты» (p. 12).

71. Помеченные ячейки — 1

Решение. Помимо случая $n = 4$, у головоломки есть решение для каждого $n > 6$, кроме $n = 9$.

Очевидно, что для $n = 1,2$ и 3 решения нет. Для $n = 4$ решение приведено в условии задачи. Для $n = 5$ решения также нет. Это утверждение можно доказать от противного. Предположим, что искомая область из пяти помеченных ячеек, у каждой из которых положительное чётное число соседей, существует. Рассмотрим самую верхнюю ячейку в самом левом столбце ячеек области. У этой ячейки

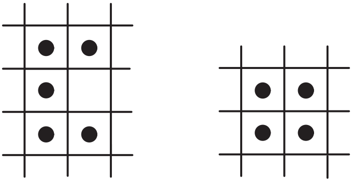


Рис. 4.43. Два возможных варианта расположения самой верхней и самой нижней ячейки в самом левом столбце с чётным количеством соседей для $n = 5$

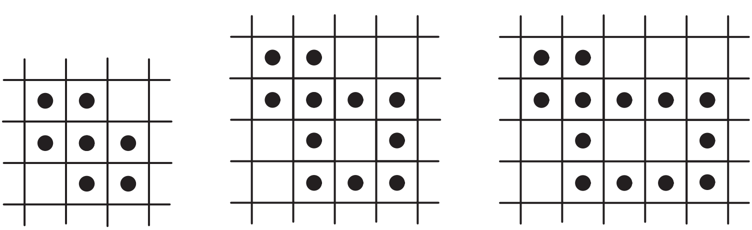
должен быть один сосед справа и один внизу. У самой нижней ячейки в самом левом столбце области должен быть один сосед справа и один сверху. Эти два возможных варианта расположения помеченных ячеек показаны на рис. 4.43.

Но область из пяти ячеек не удовлетворяет условию задачи о том, что у каждой помеченной ячейки должно быть положительное чётное количество помеченных соседних ячеек. Также невозможно пометить ещё одну ячейку в дополнение к четырём так, чтобы это условие выполнялось. Аналогичные рассуждения для $n = 6$ и $n = 9$ показывают, что решения головоломки для этих n также не существует.

Решения для $n = 7$ и $n = 11$ показаны на рис. 4.44, *а* и *б*. Второе решение можно обобщить для любого нечётного $n > 11$, просто расширяя область на две помеченные ячейки, как показано на рис. 4.44, *в* для $n = 13$.

Для любого чётного $n > 6$ решение представляет собой область в форме рамки длиной $\frac{n-2}{2}$ и высотой 3, как показано на рис. 4.45 для $n = 8$ и $n = 10$.

Комментарии. Эти решения основаны на подходе по возрастающей (стратегия «уменьшай и властвуй», применённая в обратном направлении.) Для большинства значений n существуют и другие решения. Случай с нечётным числом помеченных соседей рассмотрен в следующей головоломке «*Помеченные*



(*а*) (*б*) (*в*)

Рис. 4.44. Решение задачи «*Помеченные ячейки — 1*» при (*а*) $n = 7$, (*б*) $n = 11$ и (*в*) $n = 13$

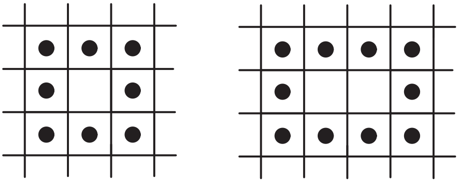


Рис. 4.45. Решение задачи «*Помеченные ячейки* — *1*» при $n = 8$ и $n = 10$

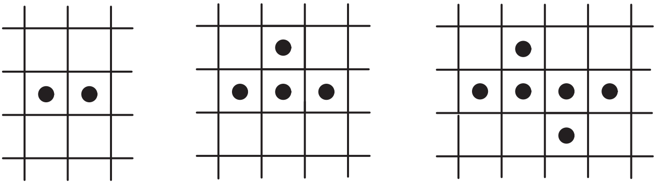
ячейки — *2*» (№ 72). Эта головоломка была включена в последнюю книгу Б. А. Кордемского [Kor05, с. 376–377].

72. Помеченные ячейки — 2

Решение. Головоломку можно решить для любого чётного числа ячеек.

При $n = 2$ очевидное решение показано на рис. 4.46, *а*. Если мы пометим две ячейки, находящиеся по соседству, например, с самой правой ячейкой в этом решении (одну по горизонтали и другую по вертикали сверху), то получим решение для $n = 4$ (рис. 4.46, *б*). Пометив горизонтальную и нижнюю соседки самой правой ячейки на рис. 4.46, *б*, получим решение для $n = 6$ (рис. 4.46, *в*). Продолжая, получаем решения для любого чётного n.

Докажем теперь, что невозможно пометить нечётное количество ячеек так, чтобы у каждой было нечётное количество помеченных соседей. Если бы это было возможно, появилось бы следующее противоречие. С одной стороны, если мы сложим число общих границ для всех помеченных ячеек, то это число должно быть чётным, поскольку каждая общая граница учитывается дважды. С другой стороны, это число должно быть нечётным, поскольку оно является суммой нечётного числа слагаемых, каждое из которых нечётно.



(*а*) (*б*) (*в*)

Рис. 4.46. Решения головоломки «*Помеченные ячейки* — *2*» при (*а*) $n = 2$, (*б*) $n = 4$ и (*в*) $n = 6$

Комментарии. Это решение для чётных n основано на подходе по возрастающей (стратегия «уменьшай и властвуй», применённая в обратном направлении). Доказательство невозможности решения для нечётных n идентично доказательству хорошо известного математического утверждения под названием «Лемма о рукопожатиях»: число людей, которые на вечеринке поздоровались за руку с нечётным числом людей, должно быть чётным (например, [Ros07, с. 599]). В нашем примере люди, пожимающие руки другим, — это помеченные ячейки, имеющие общие границы с другими помеченными ячейками. Головоломка была включена в последнюю книгу Б. А. Кордемского [Kor05, с. 376–377].

73. Погоня за петухом

Решение. Рассмотрим поле для игры как стандартную шахматную доску размером 8×8 (рис. 4.47,*а*). Петуха можно поймать, только когда петух и фермер занимают две соседние клетки (по горизонтали или вертикали) и фермер делает ход. Две соседние клетки всегда противоположных цветов. Вначале петух и фермер стоят на клетках одного цвета. Поскольку на каждом ходе они перемещаются на клетку противоположного цвета, поймать петуха невозможно, если фермер ходит первым. Если фермер ходит вторым, он всегда может загнать петуха в угол и поймать его наверняка, приближаясь к нему по диагонали. Обозначим позиции фермера и петуха как (i_F, j_F) и (i_R, j_R) соответственно, где i и j — номера клеток по горизонтали и вертикали. С геометрической точки зрения, позиция фермера (i_F, j_F) и клетки $(8, j_F)$, $(8, 8)$, $(i_F, 8)$ образуют прямоугольник, в котором находится петух и из которого он не может убежать (рис. 4.47,*б*). Этот прямоугольник на каждом ходе сужается вдоль одной стороны, пока петух не будет загнан в верхний правый угол (рис. 4.47,*в*).

Более строго определить ход фермера можно следующим образом: вычислить $i_R - i_F$, расстояние по горизонтали между клеткой (i_F, j_F), где он находится в данный момент, и клеткой, где находится петух (i_R, j_R); вычислить $j_R - j_F$, расстояние по вертикали между этими клетками, а затем определить максимум этих двух значений: $d = \max\{i_R - i_F, j_R - j_F\}$.

Фермер делает ход, чтобы уменьшить d, т. е. он движется направо, если расстояние между столбцами $j_R - j_F$ больше, чем расстояние между строками $i_R - i_F$; в противном случае он движется вверх. (После хода петуха одно расстояние всегда больше другого.)

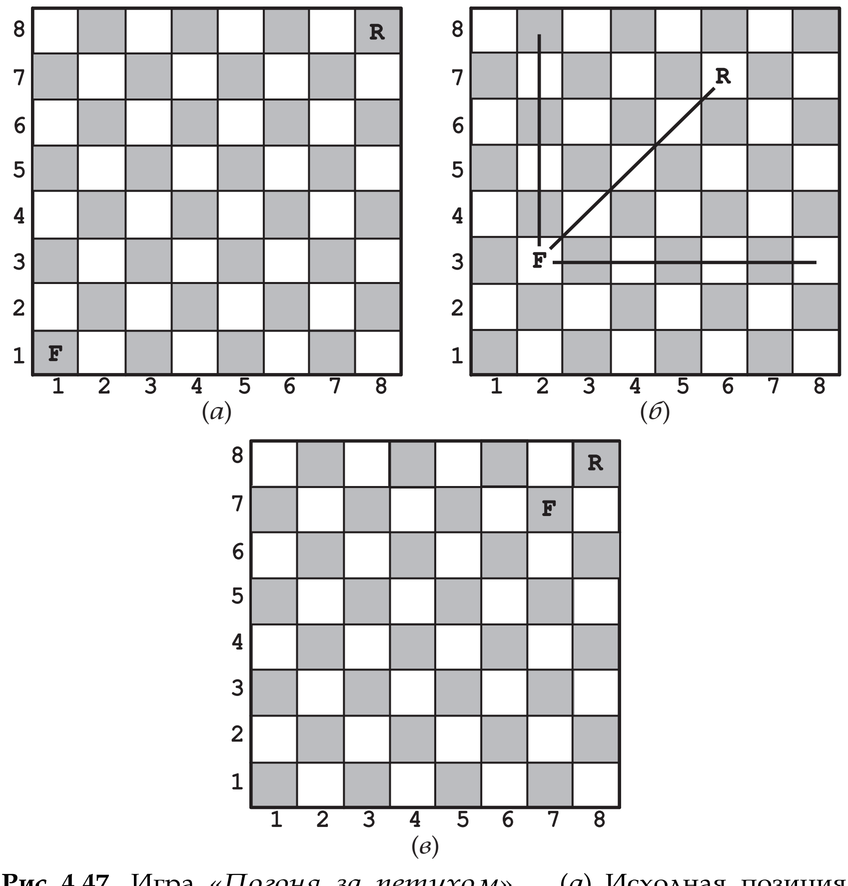


Рис. 4.47. Игра «*Погоня за петухом*». (*а*) Исходная позиция. (*б*) Выигрышная стратегия. (*в*) Позиция перед последним ходом петуха

Манхэттенское расстояние $(8 - i_F) + (8 - j_F)$ от текущей позиции фермера до верхнего правого угла уменьшается на 1 после каждого хода фермера, поэтому после 12 ходов фермер и петух будут на позиции, показанной на рис. 4.47, *в*, и на своём следующем ходе фермер петуха поймает. Таким образом, в худшем случае достаточно 14 ходов каждого участника игры (при условии, что петух ходит первым из начальной позиции на рис. 4.47, *а*). Разумеется, петух может ускорить неизбежную развязку, дойдя до квадрата, соседнего с квадратом, на котором стоит фермер, всего лишь за семь ходов.

Комментарии. В решении задачи используется идея инварианта, позволяющая определить, кто должен ходить первым, и жадная стратегия для алгоритма поимки. Можно отметить и преобразование сетки в шахматную доску, но оно в данном случае не играет решающей роли.

Задача является упрощённым вариантом головоломки «*Цыплята в огороде*», о которой говорилось во второй части учебного раздела. Похожие головоломки включены во множество сборников, например [Gar61, p. 57] и [Tan01, Problem 29.3].

74. Выбор места

Решение. Оптимальные значения для x и y — это медианы множества точек $x_1,\ldots,x_n$ и $y_1,\ldots,y_n$ соответственно.

Очевидно, что в задаче нужно минимизировать независимо друг от друга сумму расстояний по горизонтали $|x_1 - x| + \ldots + |x_n - x|$ и сумму расстояний по вертикали $|y_1 - y| + \ldots + |y_n - y|$. Таким образом, у нас есть два случая одной задачи с разными входными данными.

Чтобы минимизировать $|x_1 - x| + \ldots + |x_n - x|$, предположим, что точки $x_1,\ldots,x_n$ отсортированы в порядке неубывания. (Если это не так, то мы всегда можем сначала их отсортировать и перенумеровать.) Также воспользуемся геометрической интерпретацией модуля разности $|x_i - x|$ как расстояния между точками x_i и x на действительной прямой. Рассмотрим отдельно случаи для чётного и нечётного n.

Пусть n чётное. Сначала рассмотрим случай $n = 2$. Сумма $|x_1 - x| + |x_2 - x|$ равна $x_2 - x_1$, длине интервала с конечными точками x_1 и x_2 для любой точки x этого интервала (включая конечные точки), и больше, чем $x_2 - x_1$, для любой точки x вне этого интервала. Отсюда следует, что для любого чётного n сумма

$$\begin{aligned}
&|x_1 - x| + |x_2 - x| + \cdots + |x_{n-1} - x| + |x_n - x| = \\
&= (|x_1 - x| + |x_n - x|) + (|x_2 - x| + |x_{n-1} - x|) + \\
&\quad + \cdots + (|x - x_{\frac{n}{2}}| + |x_{\frac{n}{2}+1} - x|)
\end{aligned}$$

является минимальной, когда x принадлежит каждому интервалу с конечными точками x_1 и x_n, x_2 и $x_{n-1},\ldots, x_{\frac{n}{2}}$ и $x_{\frac{n}{2}+1}$. Поскольку каждый интервал в этой последовательности содержится в предыдущем, необходимо и достаточно, чтобы x принадлежал последнему из них. Другими словами, любое значение x (и только такое значение), для которого $x_{\frac{n}{2}} \leqslant x \leqslant x_{\frac{n}{2}+1}$, даёт решение задачи.

Пусть n нечётное. Случай $n = 1$ является тривиальным — чтобы минимизировать $|x_1 - x|$, мы должны положить $x = x_1$. Рассмотрим случай $n = 3$. Сумма

$$|x_1 - x| + |x_2 - x| + |x_3 - x| = (|x_1 - x| + |x_3 - x|) + |x_2 - x|$$

минимальна, когда $x = x_2$, поскольку это значение минимизирует как $|x_1 - x| + |x_3 - x|$, так и $|x_2 - x|$. Это рассуждение можно распространить и на любое нечётное значение n:

$$|x_1 - x| + |x_2 - x| + \dots + |x - x_{\lceil \frac{n}{2} \rceil}| + \dots + |x_{n-1} - x| + |x_n - x| =$$
$$= (|x_1 - x| + |x_n - x|) + (|x_2 - x| + |x_{n-1} - x|) + \dots + |x - x_{\lceil \frac{n}{2} \rceil}|$$

является минимальным, когда $x = x_{\lceil \frac{n}{2} \rceil}$, т. е. находится в точке, для которой количество заданных точек слева равно количеству заданных точек справа. Отметим, что средняя точка $x_{\lceil \frac{n}{2} \rceil}$, которая является $\left\lceil \frac{n}{2} \right\rceil$-й наименьшей из $x_1, \dots, x_n$, также даёт решение задачи для чётных n.

Комментарии. Для решения задачи используется стратегия «преобразуй и властвуй» — головоломка упрощается до двух случаев задачи нахождения некоторого среднего значения n данных чисел. Математики называют такое значение *медианой*; в статистике она играет важную роль. Задача эффективного нахождения медианы называется задачей выбора. У неё, разумеется, есть прямое решение: отсортировать числа в порядке неубывания и взять элемент $x_{\lceil \frac{n}{2} \rceil}$ из отсортированного списка. Более быстрые и искусные алгоритмы описаны, например, в [Lev06, p. 188–189] и [Cor09, разд. 9.2 и 9.3].

75. Инспекция бензоколонок

Решение. Согласно условию задачи предполагается посетить бензоколонку n дважды, поэтому бензоколонку $n - 1$ также нужно посетить по крайней мере дважды. Также по меньшей мере дважды нужно посетить каждую из промежуточных бензоколонок. Принимая во внимание, что бензоколонку 1 нужно посетить дважды по условиям задачи, общее число посещений будет равняться по крайней мере $2n$.

Таким образом, общая длина любого пути, удовлетворяющего условиям, будет по крайней мере $(2n-1)d$, где d — это расстояние между двумя соседними бензоколонками. Если n чётное, то путь, при котором инспектор после посещения бензоколонки с чётным номером всякий раз возвращается на одну бензоколонку назад, имеет длину, равную этой нижней грани:

$$1, 2, 1, 2, 3, 4, 3, 4, \ldots, n-1, n, n-1, n.$$

Этот путь для $n = 8$ показан на рис. 4.48.

Для нечётного n с помощью индукции докажем, что не существует пути, удовлетворяющего условию о том, что необходимо в точности дважды посетить бензоколонки $2, \ldots, n-1$. Для основного случая $n = 3$ легко удостовериться, что нужно посетить бензоколонку 2 по крайней мере три раза. Для общего случая предположим, что для нечётных $n \geqslant 3$ не существует пути, на котором промежуточные станции $2, \ldots, n-1$ посещаются в точности дважды. Докажем от противного, что пути нет также для случая $n+2$ бензоколонок. Если бы такой путь существовал, то он должен был бы заканчиваться на станции $n+2$; в противном случае нужно было бы проинспектировать станцию $n+1$ по меньшей мере три раза. Далее, обход должен был бы заканчиваться станциями $n+1$, $n+2$, $n+1$, $n+2$, и ни станцию $n+1$, ни станцию $n+2$ нельзя было бы посещать до этого. Но тогда без этих последних четырёх посещений получался бы обход, дающий такое решение задачи для n бензоколонок, при котором все промежуточные станции $2, \ldots, n-1$ посещаются дважды. Это противоречит предположению индукции и завершает доказательство.

Таким образом, для нечётных n любой путь, удовлетворяющий требованиям, должен проходить через промежуточные станции по меньшей мере трижды. Из этого следует, что для нечётных n очевидный путь

$$1, 2, \ldots, n-1, n, n-1, \ldots, 2, 1, 2, \ldots, n-1, n$$

является оптимальным.

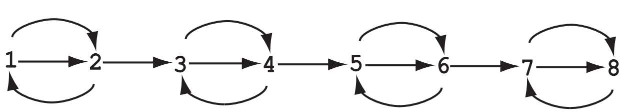


Рис. 4.48. Решение задачи «*Инспекция бензоколонок*» для $n = 8$

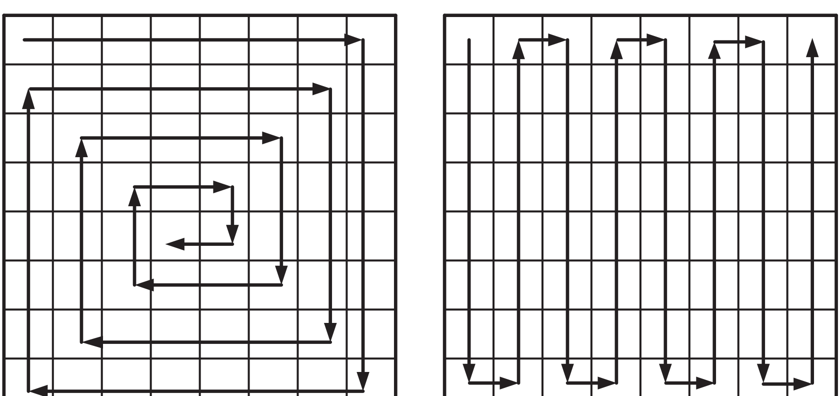
Комментарии. Головоломка является общим случаем задачи Генри Дьюдени в книге [Dud67, задача 522] и задачи Сэма Лойда в [Loy59, задача 88]. В них обеих рассмотрен случай $n = 10$.

76. Быстрая ладья

Решение. Минимальное число ходов равно $2n - 1$ для любого $n > 1$.

Оптимальный путь может начинаться в верхнем левом углу и продолжаться, следуя жадной стратегии, как можно дальше, а потом поворачивать. Итоговый путь для доски размером 8×8 показан на рис. 4.49, *а*. Число ходов при таком пути на доске $n \times n$ равно $2n - 1$.

Докажем теперь, что любой путь ладьи, который проходит по всем клеткам доски $n \times n$, где $n > 1$, состоит из по крайней мере $2n - 1$ ходов. Действительно, любой такой путь должен содержать ходы или вдоль каждого вертикального, или вдоль каждого горизонтального ряда доски. (Если путь не содержит какой-либо ход по вертикали, значит каждая клетка на этой вертикали должна быть пройдена на каком-либо ходе по горизонтали, и наоборот.) Таким образом, путь должен включать в себя или n ходов по вертикали, перемежающихся с $n - 1$ ходами по горизонтали для перехода к другой вертикали, или же n ходов по горизонтали и $n - 1$ ходов по вертикали. Это доказывает, что ладья для того, чтобы пройти



(*а*) (*б*)

Рис. 4.49. Путь ладьи с минимальным числом ходов на доске 8×8. (*а*) Жадная стратегия. (*б*) Альтернативное решение по всем клеткам доски $n \times n$, должна сделать по крайней мере $2n-1$ ходов. В свою очередь, это доказывает оптимальность описанного выше пути.

Это решение не является единственным. Одно из альтернативных решений для доски размером 8×8 показано на рис. 4.49, *б*. Разумеется, похожий путь существует для любой доски $n \times n$, где $n > 1$.

Комментарии. Первое решение основано на жадной стратегии. Как часто бывает с жадными алгоритмами, основную трудность составляет не сам алгоритм, а доказательство его оптимальности. Задача обсуждалась в книге Е. Гика «Математика на шахматной доске» [Gik76, с. 72].

77. Поиск закономерности

Решение. Если числа рассматривать как десятичные, то первые девять произведений показывают явную закономерность:

$$1 \times 1 = 1, \quad 11 \times 11 = 121, \quad 111 \times 111 = 12321,$$
$$1111 \times 1111 = 1\,234\,321, \ldots, 111\,111\,111 \times 111\,111\,111 =$$
$$= 12\,345\,678\,987\,654\,321.$$

Но потом она нарушается из-за переноса цифр в следующий разряд:

$$1\,111\,111\,111 \times 1\,111\,111\,111 = 1\,234\,567\,900\,987\,654\,321, \text{ и т. д.}$$

(Конечно же, это не исключает возможности, что какая-нибудь закономерность в итоге появится по мере увеличения количества единичек.)

Если же числа рассматривать как бинарные, то

$$\underbrace{11\ldots1}_{k} \times \underbrace{11\ldots1}_{k} = \underbrace{11\ldots1}_{k-1}\underbrace{00\ldots0}_{k}1,$$

где мы предполагаем, что 01 — это 1 для $k = 1$. Действительно, $\underbrace{11\ldots1}_{k} = 2^k - 1$, и поэтому

$$\underbrace{11\ldots1}_{k} \times \underbrace{11\ldots1}_{k} = (2^k-1)^2 = 2^{2k} - 2 \cdot 2^k + 1 = 2^{2k} + 1 - 2^{k+1}.$$

Поскольку

$$2^{2k} + 1 = 1\underbrace{00\ldots0}_{k}\underbrace{0\ldots01}_{k} \quad \text{и} \quad 2^{k+1} = 1\underbrace{00\ldots0}_{k+1},$$

мы получаем

$$2^{2k} + 1 - 2^{k+1} = \underbrace{11\ldots1}_{k-1}\underbrace{00\ldots0}_{k}1.$$

Комментарии. Десятичный вариант головоломки взят из [Ste09, p. 6].

78. Замощение прямыми тримино

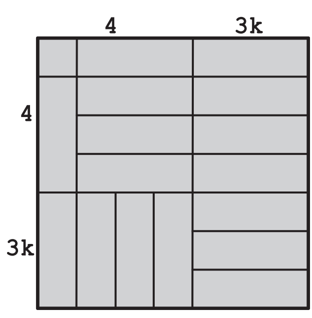
Решение. Замостить доску всегда можно.

Рассмотрим сначала случай $n \bmod 3 = 1$ и $n > 3$, т. е. $n = 4 + 3k$, где $k \geqslant 0$. Мы можем разделить квадрат на три области (обратите внимание — это идея «разделяй и властвуй» в действии!): квадрат 4×4, например в верхнем левом углу; прямоугольник $4 \times 3k$ и прямоугольник $3k \times (4+3k)$ (см. рис. 4.50, *а*). В один из углов квадрата 4×4 поместим мономино, и тогда можно замостить оставшуюся его часть. Замостить другие два прямоугольника легко (если $k > 0$), поскольку у каждого из них одна сторона равна $3k$.

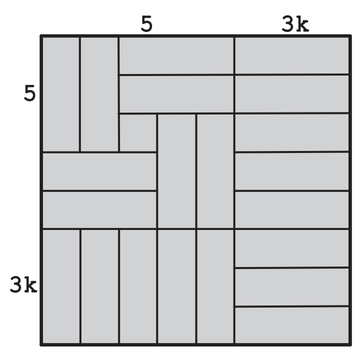
Аналогично, если $n \bmod 3 = 2$ и $n > 3$, т. е. $n = 5 + 3k$, где $k \geqslant 0$, мы можем замостить доску так, как показано на рис. 4.50, *б*.

Комментарии. Этот алгоритм замощения можно рассматривать как применение стратегии «уменьшай на три»: если квадрат размером $(n-3) \times (n-3)$ уже замощён, то замостить квадрат $n \times n$ просто.

Задача о замощении доски 8×8 прямыми тримино была представлена в статье Соломона Голомба, посвящённой замощению полимино [Gol54]. В частности, он доказал, что



(*а*)



(*б*)

Рис. 4.50. Как замостить квадрат размером $n \times n$ прямыми тримино и одним мономино при (*а*) $n = 4 + 3k$; (*б*) $n = 5 + 3k$

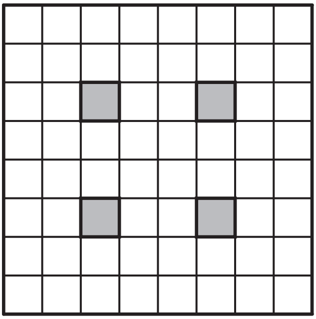


Рис. 4.51. Четыре варианта расположения мономино (показано серым) в задаче о замощении доски размером 8×8 прямыми тримино

замостить доску возможно тогда и только тогда, когда мономино помещается в одно из четырёх положений, показанных на рис. 4.51.

79. Дверцы шкафчиков

Решение. Число открытых дверей после n-го прохода равно $\lfloor\sqrt{n}\rfloor$.

Так как все дверцы вначале закрыты, то каждая дверь может быть открыта после последнего прохода тогда и только тогда, когда ключ в замке повёрнут нечётное количество раз. В дверце i $(1 \leqslant i \leqslant n)$ на проходе j $(1 \leqslant j \leqslant n)$ ключ поворачиваем тогда и только тогда, когда i делится на j; общее количество поворотов ключа в дверце i равно количеству делителей числа i. Отметим, что если j является делителем i, т. е. $i = jk$, то k тоже является делителем i. Таким образом, все делители i можно собрать попарно (например, для $i = 12$ этими парами будут 1 и 12, 2 и 6, 3 и 4); но если i является совершенным квадратом (например, $i = 16$ или $i = 4$), то парного делителя нет. Из этого следует, что у i есть нечётное число делителей тогда и только тогда, когда оно — совершенный квадрат, т. е. $i = l^2$, где l — натуральное число. Итак, после последнего обхода открыты будут только те дверцы, номера которых представляют совершенный квадрат. Общее число таких дверей, не превышающее n, равно $\lfloor\sqrt{n}\rfloor$. Эти номера являются квадратами натуральных чисел между 1 и $\lfloor\sqrt{n}\rfloor$ включительно.

Комментарии. Головоломка была опубликована в апрельском номере за 1953 год журнала *Pi Mu Epsilon Journal* (p. 330). С тех пор её включали во многие сборники головоломок как в книгах (например, [Tri85, задача 141]; [Gar88b, p. 71–72]), так и в Интернете.

80. Прогулка принца

Решение. У головоломки есть решение при любом n.

Путь принца по доске размером 8×8 показан на рис. 4.52.

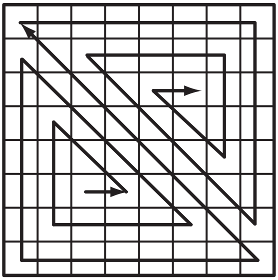


Рис. 4.52. Путь принца по доске размером 8×8

Можно сказать, что путь состоит из трёх частей: по главной диагонали доски от нижнего правого к верхнему левому углу, путь по спирали от верхнего левого угла к конечной точке, который проходит через каждую клетку выше главной диагонали в точности по одному разу, и симметричный путь по спирали, который проходит через каждую клетку ниже главной диагонали.

Легко увидеть, что такой путь можно проложить по любой доске размером $n \times n$, где $n > 1$. Для $n = 1$ головоломка имеет тривиальное решение.

Решение головоломки не является единственным. На рис. 4.53 показаны альтернативные решения для $n = 6$, 7 и 8, являющихся частными случаями для общих случаев $n = 3k$, $n = 3k + 1$ и $n = 3k + 2$.

Комментарии. Оба решения основаны на применении стратегии «разделяй и властвуй».

Идея головоломки взята из задачи о существовании возвращающегося на начальную клетку пути принца по доске 10×10 из сборника для российских математических и физических школ [Dyn71, задача 139].

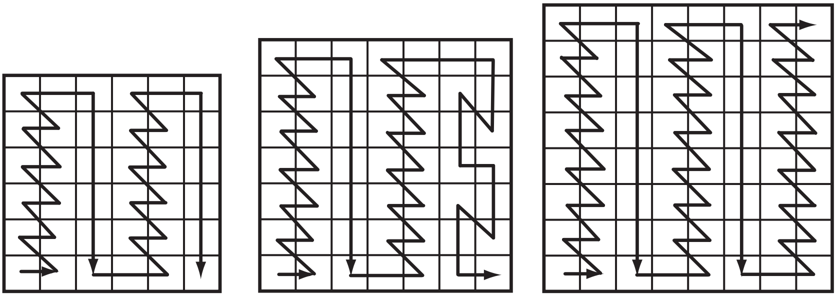


Рис. 4.53. Путь принца по доске размером $n \times n$ для $n = 3k$, $n = 3k + 1$ и $n = 3k + 2$

81. Ещё раз о задаче о знаменитости

Решение. Задачу для $n > 1$ можно решить с помощью алгоритма, задающего не более $3n - 4$ вопросов.

В отличие от более простой версии задачи, которая обсуждалась в первой части учебного раздела, мы не можем предположить, что знаменитость в группе есть. Однако та же алгоритмическая идея работает. При $n = 1$ единственный человек является знаменитостью по определению. Если $n > 1$, выберем двух человек из группы, например A и B, и спросим A, знает ли он B. Если A знает B, удалим A из группы оставшихся людей, среди которых может быть знаменитость. Если A не знает B, удалим из группы B. Решим задачу рекурсивно для оставшихся $n - 1$ человек. Если решение задачи показывает, что в группе из $n - 1$ человек нет знаменитости, то в большей группе из n человек её также не может быть, поскольку известно, что человек, удалённый из списка после первого вопроса, знаменитостью не является. Если обнаружено, что знаменитостью не является человек A или B, а, например C, то спросим, знает ли C человека, удалённого из списка после первого вопроса. Если ответ «нет», то спросим этого человека, знает ли от C. Если ответ на второй вопрос «да», то вернём C в список как знаменитость. В противном случае ответ «знаменитости нет». Если знаменитость, обнаруженная в группе из $n - 1$ человек, — это B, то спросим B, знает ли он A. Если ответ «нет», то вернём B как знаменитость. В противном случае «знаменитости нет». Если знаменитость, обнаруженная в группе из $n - 1$ человек, — это A, то спросим B, знает ли он A. Если ответ «да», то A — знаменитость. В противном случае «знаменитости нет».

Рекуррентное соотношение для числа $Q(n)$ вопросов, необходимых в худшем случае, следующее:

$$Q(n) = Q(n-1) + 3 \quad \text{при } n > 2, \quad Q(2) = 2, \quad Q(1) = 0.$$

Его можно решить методом прямой замены или обратной замены, или же с помощью формулы общего члена арифметической прогрессии. Решение следующее: $Q(n) = 2 + 3(n - 2) = 3n - 4$ для $n > 1$ и $Q(1) = 0$.

Комментарии. Этот алгоритм — прекрасный пример решения методом «уменьшай на единицу». Обсуждение алгоритма и его компьютерной реализации можно найти в книге Уди Манбера [Man89, Section 5.5]. Авторство алгоритма принадлежит С. О. Аандераа, а в статье [Kin82] описано дальнейшее улучшение алгоритма.

82. Вверх орлом

Решение. Наименьшее число ходов для решения головоломки в худшем случае равно $\left\lceil \frac{n}{2} \right\rceil$.

Мы можем представить ряд монет как чередование блоков из орлов и решек. В каждом блоке минимально может быть одна монета, а максимально n монет. Переворачивая любое количество соседних монет, мы можем уменьшить количество блоков из решек не более чем на 1, так как при переворачивании более чем одного такого блока мы переворачиваем все блоки орлов, находящиеся между ними. Таким образом, число ходов для получения нулевого количества блоков решек должно быть по меньшей мере равно количеству блоков решек в начальном ряду. Это число минимально может равняться нулю (в ряду, где все монеты расположены вверх орлом), а максимально $\left\lceil \frac{n}{2} \right\rceil$ (в ряду, где решки и орлы чередуются, а ряд начинается с решки). Алгоритм, решающий головоломку за наименьшее число ходов, может просто переворачивать все монеты в первом на текущий момент блоке решек, пока решек не останется. При самом плохом сценарии потребуется $\left\lceil \frac{n}{2} \right\rceil$ итераций.

Комментарии. Предложенный алгоритм — это применение стратегии «уменьшай на единицу». Головоломка была опубликована Л. Д. Курляндчиком и Д. Б. Фоминым в статье в журнале «Квант», посвящённой полуинвариантам [Kur89]. В качестве полуинварианта они рассматривали число пар решка-орёл и орёл-решка в ряду монет, которое не может измениться более чем на два на каждом ходу. Пример этой головоломки для 100 монет для ряда «орёл-решка…орёл-решка» был включён в книгу [Fom96, с. 194, задача 90].

83. Ханойская башня с ограничением

Решение. Головоломку можно решить за минимальное число шагов $3^n - 1$.

Если $n = 1$, переместим единственное кольцо с первого стержня сначала на средний, а затем со среднего на последний. Если $n > 1$, то:

- рекурсивно переместим верхние $n-1$ колец с первого стержня на последний через средний;
- переместим нижнее кольцо с первого стержня на средний;
- рекурсивно переместим $n-1$ колец с последнего стержня на первый через средний;
- переместим кольцо со среднего стержня на последний;
- рекурсивно переместим $n-1$ колец с первого на последний стержень через средний.

Алгоритм проиллюстрирован на рис. 4.54.

Рекуррентное соотношение для числа перемещений $M(n)$ имеет вид

$$M(n) = 3M(n-1) + 2 \quad \text{при } n > 1, \quad M(1) = 2.$$

Первые несколько значений $M(n)$ даны в таблице:

n	$M(n)$
1	2
2	8
3	26
4	80

Глядя на эти значения, можно предположить, что решением является $M(n) = 3^n - 1$; это можно подтвердить путём подстановки $M(n) = 3^n - 1$ в рекуррентное соотношение:

$$3M(n-1) + 2 = 3(3^{n-1} - 1) + 2 = 3^n - 1.$$

Рекуррентное соотношение можно решить и по-другому — с помощью стандартного метода обратных подстановок, который объясняется, например, в [Lev06, разд. 2.4].

Нетрудно доказать, что этот алгоритм решает задачу за наименьшее число шагов. Пусть $A(n)$ — это число перемещений колец, сделанных неким алгоритмом, который решает задачу. Мы докажем с помощью индукции, что

$$A(n) \geqslant 3^n - 1 \quad \text{при} \quad n \geqslant 1.$$

Для простейшего случая $n = 1$, $A(1) \geqslant 3^1 - 1$ верно. Предположим теперь, что неравенство верно для $n \geqslant 1$ колец, и рассмотрим случай для $n+1$ колец. До того как можно

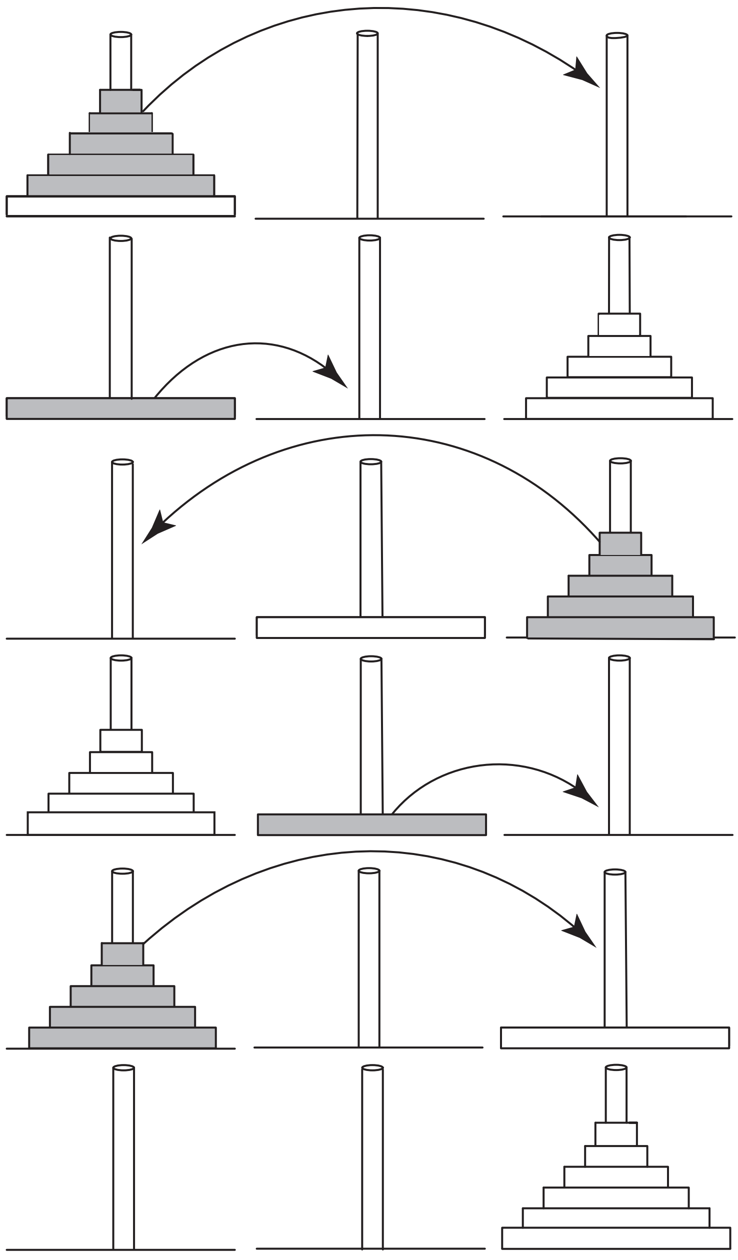


Рис. 4.54. Рекурсивный алгоритм для головоломки «*Ханойская башня с ограничением*»

будет переместить самое большое кольцо, все n меньших колец должны быть на последнем стержне. С помощью индуктивного предположения для этого потребуется по меньшей мере $3^n - 1$ перемещений колец. Перемещение самого большого кольца на средний стержень — это по крайней мере один ход. Переместить самое большое кольцо на последний стержень можно только тогда, когда все n меньших колец находятся на левом стержне. Согласно индукции, чтобы их туда переместить, нужно по меньшей мере $3^n - 1$ ходов. Перемещение самого большого кольца со среднего стержня на правый — по крайней мере один ход, а перемещение $n - 1$ колец с левого на правый — по крайней мере снова $3^n - 1$ ходов. Итого, общее число ходов алгоритма должно удовлетворять неравенству

$$A(n+1) \geqslant (3^n - 1) + 1 + (3^n - 1) + 1 + (3^n - 1) = 3^{n+1} - 1,$$

что и требовалось доказать.

Комментарии. В классической версии головоломки «*Ханойская башня с ограничением*» разрешено перемещать кольца напрямую между левым и правым стержнями. Как показано во второй части учебного раздела, это делает возможным решить задачу за минимальное число ходов $2^n - 1$. Описанный выше алгоритм решает головоломку за максимальное число ходов, при которых не повторяется конфигурация колец (см. «Ханойская башня, сложный вариант», веб-страница на [Bogom]).

Что касается используемой стратегии, то алгоритм явно основан на подходе «уменьшай на единицу». Но в отличие от стандартной версии этой стратегии, он решает три, а не два случая для размера задачи $n - 1$.

Этот вариант головоломки «*Ханойская башня с ограничением*» был опубликован в статье [Sco44] в 1944 году.

84. Сортируем блины

Решение. Задачу можно решить за $2n - 3$ переворачиваний, где $n \geqslant 2$ — число имеющихся блинов.

Стратегия «уменьшай и властвуй» ведёт к следующей схеме алгоритма. Повторяем следующий шаг, пока задача не будет решена: переместим самый большой блин, который ещё не находится в требуемом положении, на самый верх с помощью одного переворачивания, затем переместим его в конечное положение ещё одним переворачиванием. Покажем действие этой схемы более детально.

Положим k, число блинов в нижней части горки блинов, которые находятся в конечном положении, равным 0. Повторяем следующий шаг, пока k не станет равным $n-1$, т. е. пока задача не будет решена. Найдём самый большой блин над k-м снизу, который больше, чем блин непосредственно ниже его. (Если такого блина нет, задача решена.) Если этот самый большой блин лежит не на вершине горки, подведём под него лопаточку и перевернём, чтобы он оказался наверху. Начиная с $(k+1)$-го блина снизу, просмотрим горку снизу вверх и найдём первый блин, который меньше верхнего блина. Пусть это будет j-й блин снизу. (Отметим, что все блины с $(k+1)$-го до j-го включительно уже расположены в нужном порядке, поскольку верхний блин — наибольший из не отсортированных блинов.) Подведём лопаточку под j-й блин и перевернём, чтобы увеличить число блинов в нужном положении по крайней мере на 1. Затем изменим значение k, присвоив ему значение j.

Первая итерация алгоритма для случая, изображённого на рис. 2.20, показана на рис. 4.55.

Число ходов, которые делает алгоритм, в худшем случае равно $W(n) = 2n - 3$, где $n \geqslant 2$ — число блинов. (Очевидно, что $W(1) = 0$.) Эта формула вытекает из следующего рекуррентного соотношения:

$$W(n) = W(n-1) + 2 \quad \text{при } n > 2, \quad W(2) = 1.$$

Начальное условие $W(2) = 1$ является истинным, так как алгоритм решает задачу за один ход, если больший из двух блинов лежит сверху, и переворачивать не нужно, если он лежит снизу. Рассмотрим произвольную горку из $n > 2$ блинов. После двух переворачиваний или меньше алгоритм помещает самый большой блин в самый низ горки, и в дальнейших перемещениях блин уже не участвует. Таким образом, общее число переворачиваний для любой горки из n блинов ограничено сверху $W(n-1) + 2$. Эта верхняя граница для n блинов достигается следующим образом. Перевернём горку из $n-1$ блинов, блины в которой расположены наихудшим образом, и положим блин большего размера, чем все другие, непосредственно ниже верхнего блина. Для этой новой горки алгоритм сделает два хода, чтобы уменьшить задачу до переворачивания горки из $n-1$ блинов, блины в которой расположены наихудшим образом.

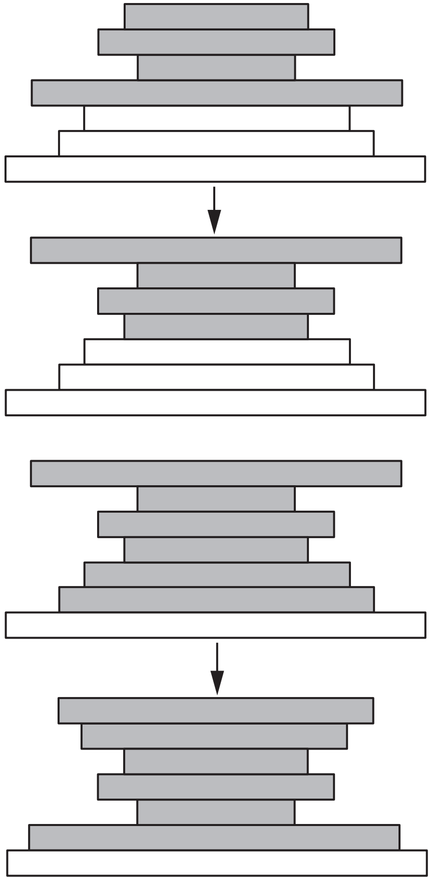


Рис. 4.55. Два переворачивания на первой итерации алгоритма «*Сортируем блины*»

Поскольку описанное рекуррентное соотношение определяет арифметическую прогрессию, получаем следующую явную формулу для её n-го члена:

$$W(n) = 1 + 2(n-2) = 2n - 3 \quad \text{при } n \geqslant 2.$$

Комментарии. Алгоритм является прекрасным примером стратегии «уменьшай и властвуй»; размер задачи в нём уменьшается нерегулярно, т. е. ни на постоянное число, ни в постоянное число раз. Однако он не является оптимальным. Минимальное число ходов при худшем сценарии лежит между $\frac{15}{14}n$ и $\frac{5}{3}n$, но точное значение не известно.

На веб-сайте Александра Богомольного [Bogom], на странице «Переворачиваем блины» можно найти апплет, иллюстрирующий головоломку. Там же есть несколько интересных фактов о ней. В частности, упоминается, что единственная исследовательская статья, опубликованная основателем *Microsoft* Биллом Гейтсом, была посвящена этой задаче.

85. Распространение сплетен — 1

Решение. Минимальное количество сообщений равно $2n-2$.

Есть несколько способов обменяться сплетнями. Например, в группе людей можно выбрать одного человека, скажем человека 1, которому каждый посылает сообщение с новостью, которую он знает. Получив все сообщения, человек 1 объединяет их сплетни со своей и посылает сообщение, содержащее все сплетни, каждому из остальных $n-1$ людей.

То же самое количество сообщений требуется следующему жадному алгоритму, который стремится максимально увеличить общее число известных сплетен после каждого посланного сообщения. Пронумеруем людей от 1 до n и пошлём первые $n-1$ сообщений таким образом: от 1 к 2, от 2 к 3 и т. д. до тех пор, пока человек n не получит сообщение от человека $n-1$ со всеми сплетнями, известными людям $1, 2, \ldots, n-1$. Далее отправим сообщение со всеми n сплетнями от человека n людям $1, 2, \ldots, n-1$.

Тот факт, что $2n-2$ сообщений — это наименьшее количество для решения головоломки, следует из того, что увеличение числа людей на одного требует по крайней мере двух дополнительных сообщений: этому человеку и от него, что и даёт данный алгоритм.

Комментарии. Эта головоломка была предложена на канадской математической олимпиаде в 1971 г. [Ton89, задача 3]. Аналогичные задачи, очевидно, имеют важное значение для специалистов по коммуникационным сетям.

86. Распространение сплетен — 2

Решение. Для $n=1$, 2 и 3 очевидными решениями являются 0, 1 и 3 шага (разговора). Вот один из нескольких алгоритмов, который достигает цели за $2n-4$ шагов при любом $n \geqslant 4$.

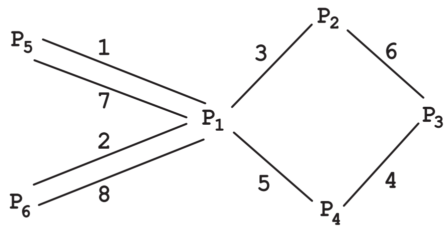


Рис. 4.56. Оптимальное распространение сплетен путём разговора двух человек при $n = 6$. Буквами в вершинах обозначены разговаривающие стороны. Последовательность разговоров показана отметками на рёбрах, соединяющих вершины

При $n=4$ нужно четыре разговора: например P_1 с P_2, P_3 с P_4, P_1 с P_4 и P_2 с P_3. При $n>4$ решение для $n=4$ может быть расширено — каждый из людей $P_5, P_6, \ldots, P_n$ сначала должен поговорить с человеком P_1, потом P_1 поговорит с P_2, P_3 с P_4, P_1 с P_4 и P_2 с P_3; а затем P_1 поговорит во второй раз с каждым из людей $P_5, P_6, \ldots, P_n$. Алгоритм для $n=6$ показан на рис. 4.56. Общее число шагов этого алгоритма равно $2(n-4)+4=2n-4$, где $n \geqslant 4$.

Комментарии. Этот алгоритм использует идею «уменьшай и властвуй», применённую в обратном порядке путём расширения решения для $n = 4$. Разработать алгоритм, решающий головоломку за $2n-4$ шагов нетрудно, но гораздо труднее доказать, что это число шагов — минимальное для $n \geqslant 4$. Доказательство и некоторые ссылки приведены в статье [Hur00]. В книге [Nie01, Problem 55] приведён другой алгоритм для решения этой задачи; предваряя головоломку, автор пишет, что ей «суждено стать классической».

87. Перевёрнутые стаканы

Решение. Для нечётных n решения нет. Если n чётное, то задачу можно решить за n ходов, что и является наименьшим требуемым числом ходов.

Если n нечётное, то $n-1$, число стаканов, переворачиваемых за один ход, является чётным. Поэтому, число стаканов, которые находятся вверх дном, всегда остаётся нечётным, каким и было изначально. Значит, конечное положение, при котором число стаканов, стоящих вверх дном, равно 0, недостижимо.

Если n чётное, задачу можно решить с помощью следующего приёма: на i-м шаге перевернём все стаканы, кроме i-го, где $i = 1, 2, \ldots, n$. (Предположим, что стаканы пронумерованы от 1 до n.)

Вот пример применения алгоритма для $n = 6$. Стаканы, стоящие вверх дном, обозначены единицами, другие обозначены нулями. Стакан, который не переворачивается на следующем ходу, выделен жирным шрифтом.

$$\mathbf{1}11111 \to 1\mathbf{0}0000 \to 00\mathbf{1}111 \to 111\mathbf{0}00 \to 0000\mathbf{1}1 \to 11111\mathbf{0} \to 000000$$

Поскольку любые два последовательных хода или не влияют на положение стаканов, или меняют положение в точности двух из них, никакой алгоритм не может решить задачу меньше чем за n ходов.

Комментарии. Головоломка представляет собой одну из нескольких версий хорошо известной задачи, в которой некие объекты могут быть в одном из двух состояний, а целью является изменить одно состояние на другое. Решение таких задач часто использует идею чётности и, если задача имеет решение, то стратегию «уменьшай и властвуй».

Головоломка была включена Чарльзом Триггом в книгу «Задачи с изюминкой» [Tri85, задача 22].

88. Жабы и лягушки

Решение. Мы можем найти число переходов и прыжков следующим образом. Единственный способ для жабы и лягушки переместиться через другую — это прыгнуть. (Неважно в данном случае, кто перепрыгивает через кого.) Таким образом, нужно n^2 прыжков для n жаб и n лягушек. Поскольку жабы не могут перепрыгивать друг через друга, первая жаба должна переместиться через $n+1$ ячеек и оказаться на ячейке $n+2$, вторая жаба должна переместиться через $n+1$ ячеек и оказаться на ячейке $n+3$, и т. д. Все жабы в сумме должны переместиться через $n(n+1)$ ячеек на свою конечную позицию. Рассуждая аналогично, лягушки тоже должны переместиться через $n(n+1)$ ячеек на конечную позицию. В сумме все они должны переместиться через $2n(n+1)$ ячеек. Поскольку один прыжок покрывает две ячейки, а прыжков n^2, то число переходов равно $2n(n+1) - 2n^2 = 2n$.

№ шага	1	2	3	4	5	Шаг
			Ячейки			
1	T	T		F	F	S_T
2	T		T	F	F	J
3	T	F	T		F	S_F
4	T	F	T	F		J_F
5	T	F		F	T	J
6		F	T	F	T	S_F
7	F		T	F	T	J_F
8	F	F	T		T	S_T
	F	F		T	T	

Рис. 4.57. Решение головоломки «*Жабы и лягушки*» при $n = 2$

№ шага	1	2	3	4	5	6	7	Шаг
			Ячейки					
1	T	T	T		F	F	F	S_T
2	T	T		T	F	F	F	J
3	T	T	F	T		F	F	S_F
4	T	T	F	T	F		F	J
5	T	T	F		F	T	F	J
6	T		F	T	F	T	F	S_T
7		T	F	T	F	T	F	J
8	F	T		4	5	6	7	J
9	F	T	F	T		T	F	J
10	F	T	F	T	F	T		S_T
11	F	T	F	T	F		T	J
12	F	T	F		F	T	T	J
13	F		F	T	F	T	T	S_F
14	F	F		T	F	T	T	J
15	F	F	F	T		T	T	S_T
	F	F	F		T	T	T	

Рис. 4.58. Решение головоломки «*Жабы и лягушки*» при $n = 3$

Для решения головоломки есть два симметричных алгоритма: один начинается с перехода последней жабы, а другой — с перехода первой лягушки. Без ограничения общности опишем первый из них. Алгоритм получаем методом «в лоб»: его ходы определяются однозначно, поскольку альтернативные ходы очевидно приводят в тупик. В частности, если есть выбор между переходом и прыжком, то нужно прыгать. Алгоритм для $n = 2$ и $n = 3$ представлен на рис. 4.57 и рис. 4.58 соответственно. Показаны состояние доски и ходы. Буква S обозначает переход, J — прыжок, подстрочные буквы обозначают, кто делает ход (T — жаба, F — лягушка). Прыжок всегда определяется однозначно, и поэтому не обозначен подстрочной буквой.

В общем случае алгоритм можно описать такой последовательностью из $2n+n^2$ букв, показывающих сделанные ходы:

$$S_{\mathrm{T}}\, J\, S_{\mathrm{F}}\, J\, J \ldots S\, \underbrace{J \ldots J}_{n-1}\, S \mid \underbrace{J \ldots J}_{n} \mid S\, \underbrace{J \ldots J}_{n-1} \ldots J\, J\, S_{\mathrm{F}}\, J\, S_{\mathrm{T}}$$

Эта последовательность — палиндром, т. е. она читается одинаково слева направо и справа налево. После ходов в левой части последовательности (слева от вертикальной черты) жабы и лягушки чередуются по схеме TFTF...TF; пустая ячейка находится после или до этой цепочки, в зависимости от того, чётное n или нечётное. Цепочка ходов в левой части состоит из n переходов S (переходы жабы и лягушки чередуются, что отражено в нижнем индексе), а между переходами находятся группы прыжков J, количество которых увеличивается от 1 до $n-1$. Центральная часть последовательности трансформирует эту схему в FTFT...FT; она состоит из n прыжков J. Правая часть последовательности завершает задачу — ходы из левой части делаются в обратном порядке.

Комментарии. Решение головоломки можно легко обобщить на случай, когда имеется m жаб и n лягушек. Тогда суммарное число ходов будет равно $mn+m+n$, из которых mn — это прыжки, а $m+n$ — это переходы. В других вариантах головоломки число пустых клеток, разделяющих жаб и лягушек, может быть больше 1.

Болл и Коксетер [Bal87, p. 124] называют источником головоломки книгу Лукаса [Luc83, p. 141–143]. Другие варианты под названиями «Овцы и козлы», «Зайцы и черепахи» можно найти в аннотированной библиографии Дэвида Сингмастера [Sin10, Section 5.R.2]. На сайте Александра Богомольного [Bogom] есть хорошая анимация головоломки и обсуждение её решения на странице «Головоломка „Жабы и лягушки“: теория и решение».

89. Перестановка фишек

Решение. Эта головоломка является двумерной версией задачи «*Жабы и лягушки*» (№ 88). Её можно решить, применив этот же алгоритм для среднего столбца. Когда алгоритм создаёт свободную клетку в строке доски в первый раз, мы можем переставлять фишки в этой строке с помощью этого же алгоритма. Таким образом, алгоритм головоломки «*Жабы и лягушки*» нужно применить в общей сложности $2n+2$ раз: по разу для каждой строки и один раз для среднего столбца. Поскольку в строке с $2n+1$ клетками алгоритм делает n^2 прыжков и $2n$ переходов, двумерный алгоритм сделает $n^2(2n+2)$ прыжков и $2n(2n+2)$ переходов, суммарно $2n(n+1)(n+2)$ ходов.

Комментарии. Очевидно, что решение основано на стратегии «преобразуй и властвуй». В принципе, упрощение двумерного варианта задачи до одномерного — это обычный метод решения задач. Такое упрощение, разумеется, возможно не всегда.

Болл и Коксетер [Bal87, p. 125] называют источником головоломки книгу Лукаса [Luc83, p. 144]. Другие ссылки можно найти в аннотированной библиографии Дэвида Сингмастера [Sin10] в разделе «Лягушки и жабы». На сайте Александра Богомольного [Bogom] есть хорошая анимация головоломки и обсуждение её решения на странице «Головоломка „Жабы и лягушки“ в размерности два».

90. Пересаживания

Решение. Пронумеруем детей от 1 до n в том порядке, в котором они сидят сначала. После этого задача сводится к получению всех перестановок чисел $1, 2, \ldots, n$ путём транспозиции двух соседних элементов.

Эти перестановки можно получить рекурсивно — сначала найти все перестановки для $1, 2, \ldots, n-1$, а затем вставить n во все возможные позиции каждой перестановки чисел $1, 2, \ldots, n-1$. Чтобы убедиться, что каждая пара последовательных перестановок отличается только транспозицией двух соседних элементов, нам нужно вставлять n в различных направлениях. Мы можем вставить n в сгенерированную перестановку либо слева направо, либо справа налево. Оказывается, что выгодно начинать вставлять n в $12\ldots(n-1)$, двигаясь справа налево, а затем менять направление всякий раз, когда нужно сделать новую перестановку $1, 2, \ldots, n-1$. Пример применения этого подхода снизу вверх для $n \leqslant 4$ показан на рис. 4.59.

Комментарии. Этот алгоритм — прекрасная иллюстрация стратегии «уменьшай на единицу». Нерекурсивная версия

Начало	1			
Вставьте 2 в перестановку 1, двигаясь слева направо	12	21		
Вставьте 3 в перестановку 12, двигаясь слева направо	123	132	312	
Вставьте 3 в перестановку 21, двигаясь слева направо	321	132	213	
Вставьте 4 в перестановку 123, двигаясь слева направо	1234	1243	1423	4123
Вставьте 4 в перестановку 132, двигаясь слева направо	4132	1432	1342	1324
Вставьте 4 в перестановку 312, двигаясь слева направо	3124	3142	3412	4312
Вставьте 4 в перестановку 321, двигаясь слева направо	4321	3421	3241	3214
Вставьте 4 в перестановку 231, двигаясь слева направо	2314	2341	2431	4231
Вставьте 4 в перестановку 213, двигаясь слева направо	4213	2413	2143	2134

Рис. 4.59. Генерация перестановок снизу вверх

алгоритма известна в информатике как алгоритм Джонсона— Троттера, по имени двух учёных, которые независимо друг от друга опубликовали её около 1962 г.

Как пишет Мартин Гарднер [Gar88b, p. 74], на самом деле алгоритм был придуман польским математиком Гуго Штейнгаузом для решения задачи абака [Ste64, задача 98]. Задача иногда называется «Задачей Лемера о мотеле», по названию статьи Д. Г. Лемера [Leh65], который рассматривал более общую задачу о перестановке чисел, не все из которых могут быть различными.

91. Горизонтальные и вертикальные домино

Решение. Замостить доску возможно тогда и только тогда, когда n делится на 4.

Если n нечётное, то у доски $n \times n$ нечётное число квадратов. Поэтому замостить такую доску плитками домино невозможно, потому что любое количество доминошек покрывает чётное число квадратов.

Если n делится на 4, т. е. $n = 4k$, доску можно разделить на $4k^2$ квадратов размером 2×2. Поскольку $4k^2$ — это чётное число, можно замостить половину квадратов 2×2 горизонтальными домино, а другую половину — вертикальными, и таким образом получить решение задачи.

Если n чётное, но не делится на 4, т. е. $n = 2m$, где m нечётное, замостить доску $n \times n$ одинаковым числом горизонтальных и вертикальных домино невозможно. Для доказательства раскрасим строки доски чередующимися цветами (см. для примера рис. 4.60). Обратите внимание, что любая горизонтальная плитка домино покроет две клетки одного цвета, а любая вертикальная — две клетки разных цветов. Мы

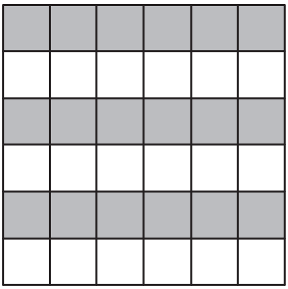


Рис. 4.60. Раскрашивание доски $n \times n$ для случая, когда n чётное, но не делится на 4

хотим замостить клетки доски, число которых $n^2 = 4m^2$, с помощью t горизонтальных плиток домино и t вертикальных, где $t = m^2$. У раскрашенной доски $2m^2$ клеток одного цвета и $2m^2$ другого. Вертикальные домино должны покрыть m^2 клеток каждого цвета, а оставшиеся m^2 клеток нужно замостить горизонтальными домино. Но это невозможно сделать ни для одного из этих двух цветов, поскольку m^2 нечётное, а число покрытых домино квадратов должно быть чётным.

Комментарии. Решение типично для задач о замощении плиткой. Когда это возможно, доску разделяют на области, которые легко замостить. Когда невозможно, используется инвариант, часто основанный либо на чётности, либо на раскрашивании. Примеры таких задач приведены во второй части учебного раздела.

Эта головоломка о замощении плиткой хорошо известна: например, похожая задача есть в книге [Eng99, p. 26, Problem 9].

92. Замощение трапециями

Решение. У головоломки есть решение тогда и только тогда, когда n не делится на 3.

Если мы подсчитаем общее количество $T(n)$ маленьких треугольников по слоям, начиная с основания большого треугольника (рис. 4.61), то получим следующую формулу:

$$T(n) = [n + 2(n-1) + 2(n-2) + \cdots + 2 \cdot 1] - 1 =$$

$$= n + \frac{2(n-1)n}{2} - 1 = n^2 - 1.$$

Поскольку одна трапеция состоит из трёх маленьких треугольников, задача имеет решение только тогда, когда $n^2 - 1$

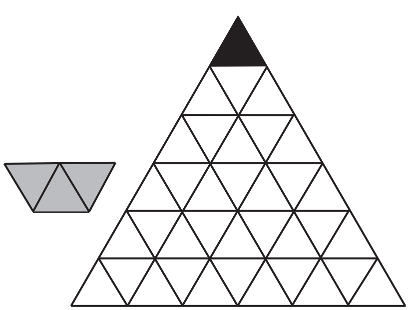
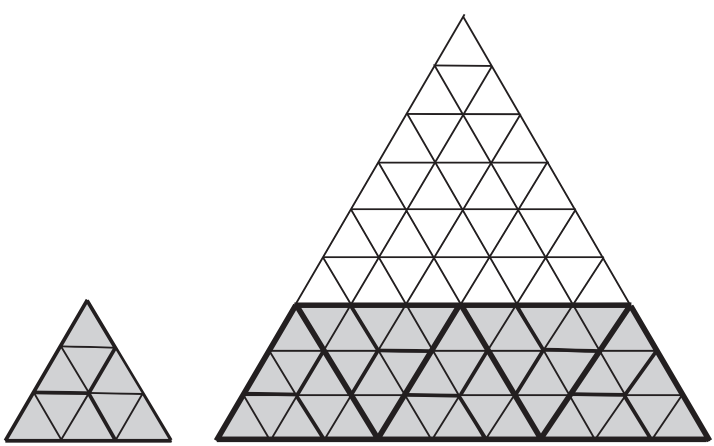


Рис. 4.61. Область, которую нужно замостить плитками в виде трапеций (закрашена серым) для $n = 6$

делится на 3. Но n^2-1 делится на 3 тогда и только тогда, когда n не делится на три. Это можно показать, рассмотрев случаи $n=3k$, $n=3k+1$ и $n=3k+2$.

Мы покажем, что это условие является не только необходимым, но и достаточным для того, чтобы задача имела решение. Но предварительно докажем, что если $n=3k$ и ни один маленький треугольник не удалён из рассматриваемой области, то её можно замостить трапециями. Это легко доказать методом индукции по k. При $k=1$ область можно замостить тремя трапециями (рис. 4.62, *а*). Докажем, что если область можно замостить трапециями при $n=3k$, где $k \geqslant 1$, то это возможно и при $n=3(k+1)$. Рассмотрим отрезок, параллельный основанию большого треугольника и проходящий через точки, делящие его стороны в отношении $3:3k$ (рис. 4.62, *б*). Этот отрезок делит нашу область на трапецию (ниже отрезка) и равносторонний треугольник (выше отрезка). Трапецию можно разделить на $(k+1)+k$ равносторонних треугольников с длиной стороны 3 и поэтому её можно замостить трапециями. Равносторонний треугольник можно замостить, пользуясь предположением индукции.

Теперь рассмотрим случай равностороннего треугольника с длиной стороны $n=3k+1$, у которого отрезан самый



(*а*) (*б*)

Рис. 4.62. (*а*) Замощение трапециями всей треугольной области при $n=3$. (*б*) Рекурсивное замощение всей треугольной области при $n=3k$, $k>1$

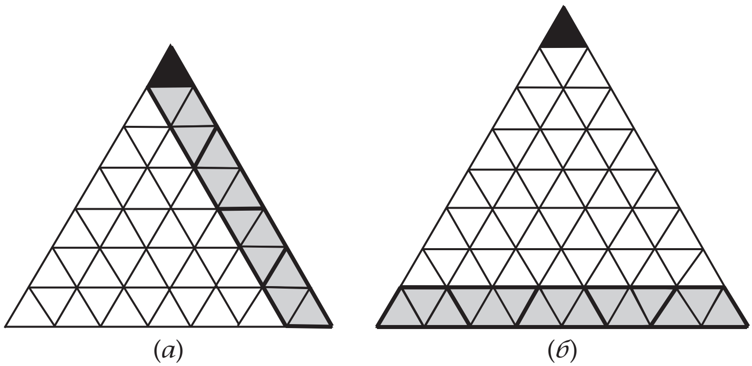


Рис. 4.63. Первый шаг в замощении треугольной области для (*а*) $n = 7$ и (*б*) $n = 8$

верхний равносторонний треугольник. Если мы разместим $2k$ трапеций вдоль одной из сторон области (см., например, рис. 4.63, *а*), то задача сведётся к замощению всего равностороннего треугольника с длиной стороны $3k$, что можно сделать, как показано выше. И если $n = 3k + 2$, мы можем разместить $2k+1$ трапеций вдоль основания большого треугольника, уменьшив тем самым размер задачи до $n = 3k + 1$ (см. рис. 4.63, *б* в качестве примера).

Комментарии. Решение задачи использует идею инварианта, а также стратегии «уменьшай и властвуй» и «преобразуй и властвуй».

Интересно, что для $n = 2k$ головоломку можно также решить с помощью такого алгоритма типа «разделяй и властвуй». Если $n = 2$, то наша область конгруэнтна одной плиточке в форме трапеции. Если $n = 2k$, где $k > 1$, расположим более длинное основание первой плитки в середине основания данной области. Затем нарисуем три прямые линии, соединяющие середины трёх сторон треугольника, из которого получается наша область. Этим мы рассечём область на четыре конгруэнтные части, подобные изначальной области, но в точности в половину её размера (рис. 4.64, *а*). Итак, каждую часть можно замостить с помощью того же алгоритма, т. е. рекурсивно. Этот алгоритм показан на рис. 4.64, *б* при $n = 8$. Очевидно, он схож с алгоритмом для замощения плитками тримино области размером $2n \times 2n$, в которой нет

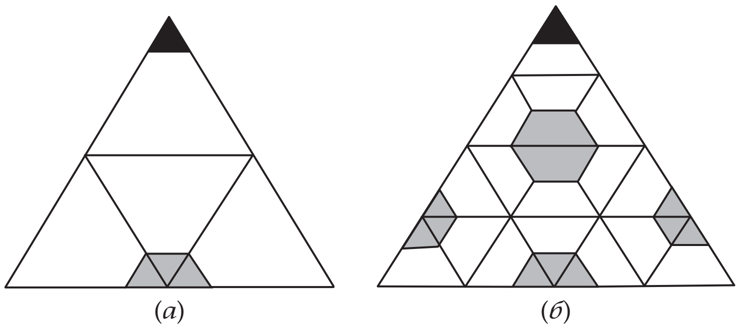


Рис. 4.64. (*а*) Разделение треугольной области на четыре одинаковые части размером в половину исходной. (*б*) Покрытие плитками с помощью стратегии «разделяй и властвуй» для $n = 8$

одной клетки (эта задача была рассмотрена в первой части учебного раздела).

Вариант головоломки для $n = 2k$ был использован Рональдом Бэкхаузом как пример задачи на применение индукции в курсе решения алгоритмических задач в Ноттингемском университете [Backh].

93. Стрельба по линкору

Решение. Минимальное количество выстрелов, необходимых для того, чтобы гарантировано подбить линкор (прямоугольник 4×1 или 1×4), — 24. Одно из возможных решений показано на рис. 4.65.

Меньше чем 24 выстрела может оказаться недостаточно. Это проиллюстрировано на рис. 4.66, где отмечены 24 возможных положения линкора; чтобы подбить каждый линкор, нужен один выстрел.

Комментарии. Головоломка иллюстрирует анализ худшего сценария в достаточно необычном применении. В статье [Gik80] журнала «Квант» приведено другое решение, дающее тот же ответ — двадцать четыре выстрела; оно показано на рис. 4.67.

Решение этой же задачи для доски 8×8 (необходимо сделать минимально 21 выстрел) было дано ранее Соломоном Голомбом в первом издании книги о полимино [Gol94].

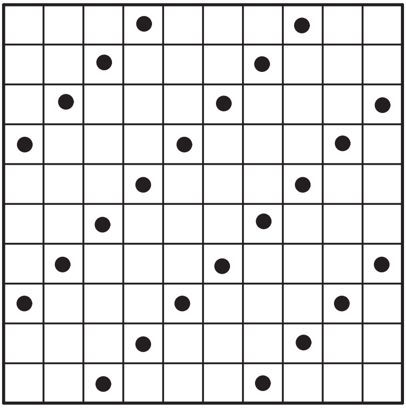


Рис. 4.65. Решение головоломки «*Стрельба по линкору*»

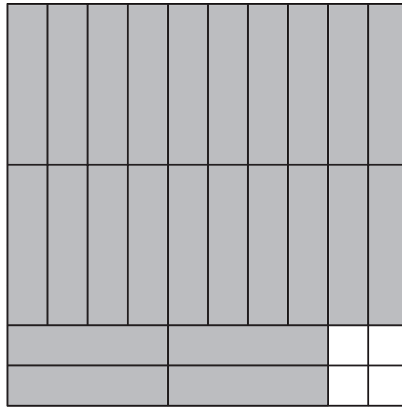


Рис. 4.66. Двадцать четыре возможных расположений линкора

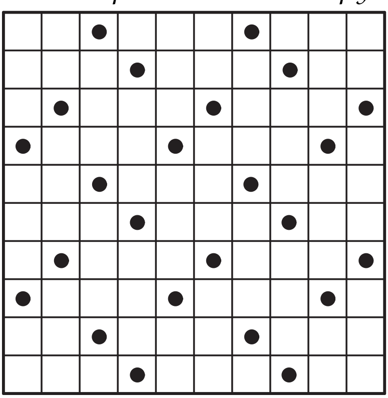


Рис. 4.67. Альтернативное решение головоломки «*Стрельба по линкору*»

94. Поиск в отсортированном массиве

Решение. Сначала перевернём карту в верхнем правом углу массива и сравним номер на ней с заданным номером. Если номера совпадают, задача решена. Если нужный номер меньше, чем номер на карте, то нужный номер не может находиться в последнем столбце, и мы переворачиваем карту слева в предыдущем столбце. Если нужный номер больше, чем номер на карте, то он не может находиться в первой строке, и мы переходим вниз к карте в следующей строке. Повторяем этот шаг, пока не найдём нужный номер, или же пока поиск не выйдет за пределы массива.

В любом случае задача решена. Последовательность переворачиваемых карт образует зигзаг, идущий влево-вниз от верхнего правого угла к какой-либо карте массива. Самый длинный путь заканчивается в нижнем левом углу, после переворачивания 19 карт. Длиннее путь быть не может, поскольку в нём не более девяти горизонтальных и девяти вертикальных сегментов.

Комментарии. Алгоритм основан на стратегии «уменьшай на единицу», поскольку на каждой итерации он уменьшает или число строк, или число столбцов, в которых может находиться заданный номер.

Головоломка появилась в нескольких публикациях, посвящённых собеседованиям по техническим специальностям, как в печатном виде, так и в сети (см., например, [Laa10, Problem 9.6]).

95. Максимальный и минимальный вес

Решение. Разделим предметы на $\left\lfloor \frac{n}{2} \right\rfloor$ пар; если n нечётное, то отложим один предмет в сторону.

Взвесим каждую пару предметов, чтобы определить какой легче, а какой тяжелее. (Если они весят одинаково, то можно выбрать любой.) За $\left\lfloor \frac{n}{2} \right\rfloor - 1$ взвешиваний найдём самый лёгкий из $\left\lfloor \frac{n}{2} \right\rfloor$ более лёгких предметов, а затем самый тяжёлый из $\left\lfloor \frac{n}{2} \right\rfloor$ более тяжёлых. Если n чётное, задача решена; если n нечётное, сравним вес отложенного предмета с уже найденными самым лёгким и самым тяжёлым и определим самый лёгкий и самый тяжёлый из всех предметов.

Общее число взвешиваний в алгоритме $W(n)$ определяется следующими формулами. Для чётного n

$$W(n) = \frac{n}{2} + 2\left(\frac{n}{2} - 1\right) = \frac{3n}{2} - 2.$$

Для нечётного n

$$W(n) = \left\lfloor \frac{n}{2} \right\rfloor + 2(\left\lfloor \frac{n}{2} \right\rfloor - 1) + 2 = 3\left\lfloor \frac{n}{2} \right\rfloor = 3\,\frac{n-1}{2} = \frac{3n}{2} - \frac{3}{2}.$$

Легко проверить, что обе формулы для случаев чётного и нечётного n сводятся к одной:

$$W(n) = \left\lceil \frac{3n}{2} \right\rceil - 2.$$

Действительно, если n чётное, то $\left\lceil \frac{3n}{2} \right\rceil - 2 = \frac{3n}{2} - 2$. Если $n = 2k + 1$ нечётное, то $\left\lceil \frac{3n}{2} \right\rceil - 2 = \left\lceil \frac{3(2k+1)}{2} \right\rceil - 2 = \left\lceil 3k + \frac{3}{2} \right\rceil - 2 = = 3k = 3\frac{n-1}{2} = \frac{3n}{2} - \frac{3}{2}$.

Комментарии. Тот же по существу алгоритм может быть получен с помощью применения стратегии «разделяй и властвуй»: разделим предметы на две равные (или почти равные группы), найдём самый лёгкий и самый тяжёлый предмет в каждой группе, а затем сравним вес двух самых лёгких и двух самых тяжёлых.

Эта задача широко известна в информатике, обычно она формулируется как задача о нахождении самого большого и самого маленького элемента во множестве из n чисел. Доказано, что $\left\lceil \frac{3n}{2} \right\rceil - 2$ — это минимальное количество сравнений, необходимых для решения задачи с помощью любого алгоритма сравнения в наихудшем случае (см. [Poh72]).

96. Замощение лестницы

Решение. Для случая S_2 ответ очевиден — замостить одним тримино. Для $n > 2$ замощение возможно тогда и только тогда, когда или $n = 3k$, где $k > 1$, или $n = 3k + 2$, где $k > 1$.

Очевидно, что S_n можно замостить тримино, если общее число квадратиков в S_n делится на 3. Общее число квадратиков в S_n равно n-му треугольному числу

$$T_n = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}.$$

Если $n = 3k$, где k чётное (т. е. $k = 2m$), то $T_n = \frac{6m(6m+1)}{2} = = 3m(6m+1)$ делится на 3. Если $n = 3k$, где k нечётное (т. е. $k = 2m + 1$), то $T_n = \frac{(6m+3)(6m+4)}{2} = 3(2m+1)(3m+2)$ также делится на 3. Аналогично можно проверить, что если $n = 3k + 1$, где k чётное или нечётное, T_n не делится на 3. И наконец, можно показать, что если $n = 3k + 2$, то T_n делится на 3 как для чётных, так и для нечётных значений k.

Единственно возможный вариант покрытия первой ступеньки в лестнице S_3 и последней ступеньки в S_5 (рис. 4.68) показывает, что эти лестницы нельзя замостить угловыми тримино.

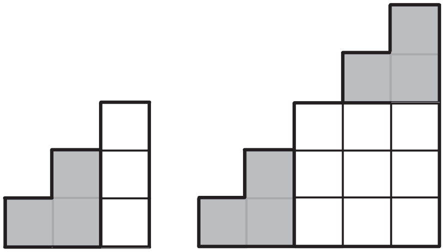


Рис. 4.68. Покрыть «лестницы» S_3 и S_5 невозможно

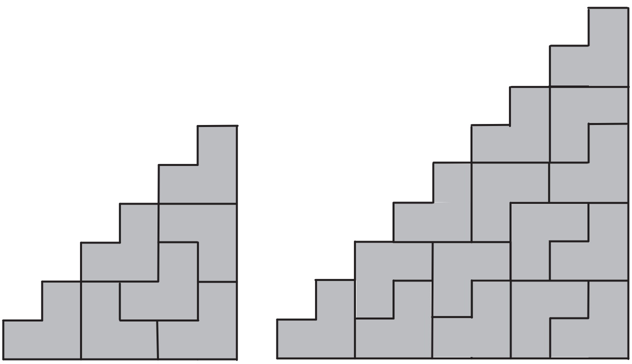


Рис. 4.69. Замощение S_6 и S_9

Теперь покажем, что любую лестницу S_n, где $n = 3k$, $k > 1$, можно замостить угловыми тримино с помощью следующего рекурсивного алгоритма. Замощение S_6 и S_9 показано на рис. 4.69.

Если $n = 3k$, где k — чётное натуральное число, большее 2 (т. е. $n = 6m = 6+6(m-1)$, где $m > 1$), то область в форме лестницы S_n можно разделить на лестницы S_6 и $S_{6(m-1)}$ и прямоугольник размером $6 \times 6(m-1)$. Замощение S_6 показано на рис. 4.69, $S_{6(m-1)}$ можно замостить рекурсивно, а прямоугольник можно замостить, разделив его на прямоугольники размером 3×2, каждый из которых покрывается двумя тримино (рис. 4.70, *а*).

Если $n = 3k$, где k — нечётное натуральное число, большее 3 (т. е. $n = 6m+3 = 9+6(m-1)$, где $m > 1$), то область S_n можно разделить на области в форме лестницы S_9 и $S_{6(m-1)}$ и прямоугольник $9 \times 6(m-1)$. Замощение S_9 показано на рис. 4.69;

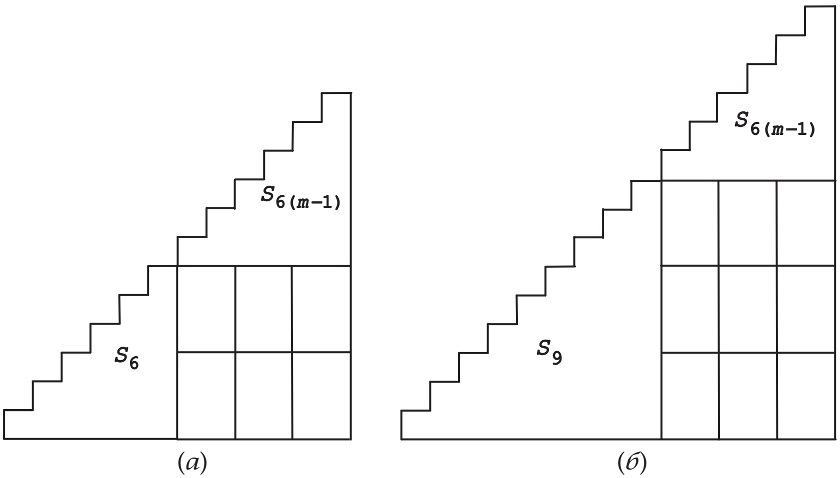


Рис. 4.70. Замощение (*а*) $S_{6+6(m-1)}$ и (*б*) $S_{9+6(m-1)}$

$S_{6(m-1)}$ можно замостить, как показано выше, а прямоугольник можно поделить на прямоугольники размером 3×2, каждый из которых покрывается двумя тримино (рис. 4.70, *б*).

Таким образом, мы получили алгоритм для замощения любой области в формы лестницы S_n, где $n = 3k$, $k > 1$.

Наконец покажем, что любую область S_n, где $n = 3k + 2$, $k > 1$, можно покрыть угловыми тримино следующим образом. Мы можем разделить S_n на S_2, S_{3k} и прямоугольник размером $2 \times 3k$ (рис. 4.71). Область S_2 можно покрыть единственным тримино, S_{3k} можно замостить с помощью описанного выше алгоритма, а прямоугольник — разделив его на прямоугольники 2×3, каждый из которых покрывается двумя тримино.

Комментарии. Для решения задачи используются несколько приёмов разработки и анализа алгоритмов: формула треугольных чисел, инвариант ($T_n \bmod 3 = 0$), стратегии «разделяй и властвуй» (разделение области) и «уменьшай (на 6) и властвуй».

Случай головоломки для $n = 8$ был включён в книгу А. Спивака [Spi02, задача 80].

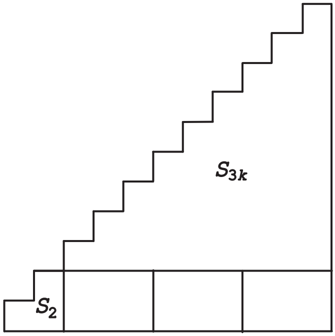


Рис. 4.71. Замощение области S_{3k+2}

97. Обмен в колоде карт

Решение. Игра всегда заканчивается после конечного числа итераций.

Действительно, король может появиться на верху колоды не более одного раза, так как после этого алгоритм переместит его на последнюю, тринадцатую, позицию и никакая другая карта (поскольку она младше) не сдвинет его оттуда. Аналогично королева может оказаться на верху колоды не более двух раз: после того, как она окажется там в первый раз и будет перемещена алгоритмом на 12-ю позицию, только король сдвинет её оттуда, а это может произойти только один раз. Валет может оказаться на верху колоды не боле четырёх раз: после первого раза его могут сдвинуть с окончательной позиции только король или королева, которые могут оказаться сверху $1 + 2 = 3$ раза. В общем случае, что можно доказать с помощью индукции, карта со значением $i, 2 \leqslant i \leqslant 13$, может появиться на верху колоды не более чем $1 + (1 + 2 + \ldots + 2^{12-i}) = 2^{13-i}$ раз, где $(1 + 2 + \ldots + 2^{12-i}) = 2^{13-i} - 1$ является верхней границей числа раз, когда карта со значением больше i может появиться на верху. В частности, туз окажется на верху (и тогда игра закончится!) не позже чем после $2^{12} - 1$ появлений на верху всех остальных карт. На самом деле, в самой длинной игре 80 ходов, что было установлено с помощью компьютерной программы [Knu11, p. 721].

Комментарии. Эта задача — пример алгоритмической головоломки, целью которой является доказать, что повторяющиеся операции останавливаются после конечного числа итераций при любых входных данных.

Оригинальное название игры «The Game of Topswops» было придумано её изобретателем — Джоном Х. Конвеем, английским математиком, который с 1986 года работает в Принстонском университете США [Gar88b, p. 76]. Разумеется, можно играть и с бóльшим количеством карт $n \geqslant 1$, написав на них значения от 1 до n.

98. Ромбопалиндром

Решение. Ответ 63 504.

Как мы упомянули в подсказке, сначала подсчитаем число способов прочитать CAT I SAW. Любая такая последовательность начинается с буквы C в центре и находится в одном из четырёх треугольников, образованных диагоналями ромба. Один из этих треугольников показан на рис. 4.72.

Число последовательностей, дающих фразу CAT I SAW в треугольнике, можно найти с помощью стандартного применения динамического программирования (см. первую часть учебного раздела). Это число можно подсчитать по диагоналям, параллельным гипотенузе треугольника, суммируя соседние числа слева и ниже нужных букв. Эти числа формируют треугольник Паскаля. Сумма этих чисел на гипотенузе треугольника — границе ромба — равна 2^6.

Общее число последовательностей, дающих CAT I SAW для всего ромба, затем получается по формуле $4 \cdot 2^6 - 4$. (Нам нужно вычесть 4, чтобы компенсировать одинаковые последовательности вдоль диагоналей ромба.) Итак, общее число способов прочесть WAS IT A CAT I SAW определяется формулой $(4 \cdot 2^6 - 4)^2 = 63\,504$.

Комментарии. Для решения задачи помимо динамического программирования дважды используется симметричность:

```
            W
          W A W
        W A S A W
      W A S I S A W
    W A S I T I S A W
  W A S I T A T I S A W
W A S I T A C A T I S A W
  W A S I T A T I S A W
    W A S I T I S A W
      W A S I S A W
        W A S A W
          W A W
            W
```

W_1
$A_1 W_6$
$S_1 A_5 W_{15}$
$I_1 S_4 A_{10} W_{20}$
$T_1 I_3 S_6 A_{10} W_{15}$
$A_1 T_2 I_3 S_4 A_5 W_6$
$C_1 A_1 T_1 I_1 S_1 A_1 W_1$

Рис. 4.72. Ромб из букв и число путей в треугольнике от буквы C к каждой букве во фразе CAT I SAW подсчитывается, сколько раз половина палиндрома встречается в четверти заданной области.

Головоломка взята из книги Сэма Лойда [Loy59, задача 109]; она также была включена в книгу Дьюдени «Кентерберийские головоломки» [Dud02, задача 30] (только там была буква R, а не C). Дьюдени вывел общую формулу для числа способов прочесть такой палиндром из $2n+1$ букв ($n>0$), расположенных в форме ромба: $(4\cdot 2^n-4)^2$.

99. Обратная сортировка

Решение. Головоломку можно решить за $\frac{(n-1)^2}{4}$ обменов карт, что является минимальным для любого нечётного n; для чётного n решения нет.

При любой допустимой последовательности перестановок карт две карты, которые мы меняем местами, находятся обе либо в чётных позициях, либо в нечётных. Поэтому, если n чётное, то первую карту с самым большим номером нельзя переместить на последнюю позицию, которая чётная.

Если n нечётное ($n=2k-1$, $k>0$), то задачу можно решить с помощью алгоритма сортировки, такого как сортировка пузырьком или сортировка вставкой, применив его сначала к номерам в нечётных позициях, а затем к номерам в чётных позициях. Оба этих алгоритма работают, меняя неупорядоченные пары соседних элементов. Например, если сортировка пузырьком применяется к нечётным номерам, она меняет первый номер с третьим, затем третий с пятым, и т. д., пока самый большой номер не окажется в последней позиции. Затем, на втором проходе, она отодвинет второй по старшинству номер в нечётной позиции к его последней позиции, и т. д. После $k-1$ шагов номера в нечётных позициях будут отсортированы.

Сортировка пузырьком делает $\frac{(s-1)s}{2}$ перестановок в строго уменьшающемся массиве размером s. Таким образом, она сделает $\frac{(k-1)k}{2}$ перестановок карт на нечётных позициях и $\frac{(k-2)(k-1)}{2}$ перестановок карт на чётных позициях, что в целом даст

$$\frac{(k-1)k}{2}+\frac{(k-2)(k-1)}{2}=(k-1)^2=\frac{(n-1)^2}{4}.$$

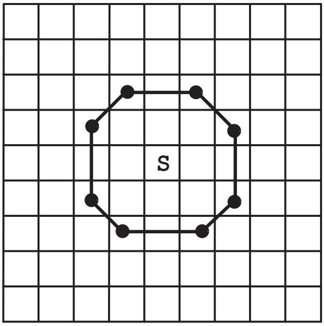
Число перестановок, осуществляемых любым алгоритмом, не может быть меньше по следующей причине. Нам разрешено только менять местами соседние элементы в двух отсортированных в обратном порядке последовательностях карт в позициях с нечётными номерами и в позициях с чётными номерами. Эта операция уменьшает общее число инверсий — двух элементов вне порядка — на одну в их последовательности. Общее число инверсий в строго уменьшающейся последовательности из s элементов равно $\frac{(s-1)s}{2}$: первый элемент больше всех $s-1$ элементов, следующих за ним; второй больше, чем $s-2$ элементов, и т. д. В сумме это и даёт $(s-1)+(s-2)+\ldots+1=\frac{(s-1)s}{2}$ инверсий. Таким образом, это число и является минимальным числом перестановок, необходимым при любой последовательности операций, уменьшающих число инверсий на единицу за раз.

Комментарии. Главные идеи в этой головоломке — использование принципа чётности и инверсия. Головоломка — усложнение задачи 155 в [Dyn71], где рассматривался только случай для $n=100$.

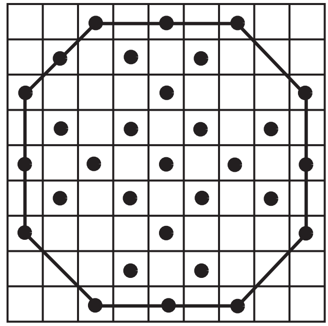
100. Куда доскачет конь

Решение. Ответом является число $7n^2+4n+1$ для $n \geqslant 3$, и 8 и 33 для $n=1$ и 2 соответственно.

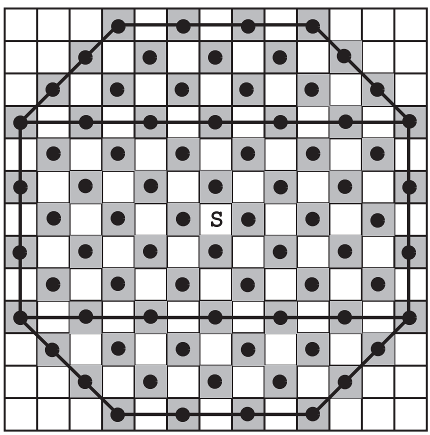
Поля, на которых конь может оказаться за $n=1$, 2 и 3 ходов, показаны на рис. 4.73. Как видно из этого рисунка, число $R(n)$ различных полей, на которых конь может оказаться за n шагов, для $n=1$, 2 и 3 равно $R(1)=8$, $R(2)=33$ и $R(3)=76$. Пользуясь математической индукцией, несложно показать, что для любого нечётного $n \geqslant 3$ все поля, достижимые за n шагов, суть поля цвета, противоположного цвету начального поля, и лежащие на границе или внутри восьмиугольника с центром в начальном поле с горизонтальными и вертикальными сторонами длиной в $2n+1$ полей (см. рис. 4.73,*в* для $n=3$). Для любого чётного $n \geqslant 3$ всё точно так же, с той лишь разницей, что цвет у достижимых полей тот же, что у начального. Чтобы получить формулу для $R(n)$, где $n \geqslant 3$, можно, например, разделить восьмиугольник на прямоугольник $(2n+1)\times(4n+1)$ в центре и две трапеции сверху и снизу от прямоугольника. Прямоугольник состоит из $n+1$ строк, в каждой из которых $2n+1$ достижимых полей, вперемешку с n строками, в каждой из которых $2n$ достижимых полей. Число достижимых полей



(*а*)



(*б*)



(*в*)

Рис. 4.73. Поля, до которых конь может дойти из поля S за (*а*) один ход, (*б*) два хода и (*в*) три хода

в трапециях можно получить, пользуясь стандартной формулой для суммы членов арифметической прогрессии:

$$2[(n+1)+(n+2)+\cdots+2n] = 2\,\frac{n+1+2n}{2}\,n = (3n+1)n.$$

Отсюда сразу получаем формулу для общего числа достижимых полей для $n \geqslant 3$:

$$R(n) = (2n+1)(n+1)+2n^2+(3n+1)n = 7n^2+4n+1.$$

Комментарии. Эта задача была включена в книгу Е. Гика «Математика на шахматной доске» [Gik76, с. 48–49].

101. Перекраска комнат

Решение. Алгоритм, изображённый на рис. 4.74, потребует перекраски $2(1+5+9+13)=56$ комнат.

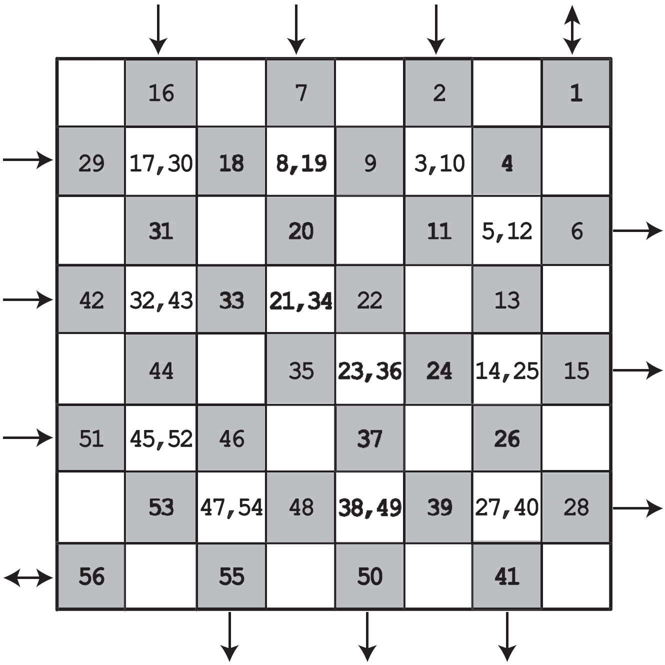


Рис. 4.74. Решение задачи «*Перекраска комнат*»: числа обозначают, в каком порядке следует красить комнаты

Комментарии. В решении используется симметричность дворца относительно главной диагонали, так что его можно рассматривать как пример использования стратегии «разделяй и властвуй». Решение авторам прислал Норберт Нолте из Германии.

102. Обезьянка и кокосовые орехи

Решение. Ответом является число 15 621.

Пусть n — начальное число кокосов, a, b, c, d и e — число кокосов, которые взял первый, второй, третий, четвёртый и пятый моряки соответственно, и пусть f — число кокосов, которое каждый из них получил утром. Тогда получается следующая система уравнений:

$$n = 5a + 1,$$
$$4a = 5b + 1,$$
$$4b = 5c + 1,$$
$$4c = 5d + 1,$$
$$4d = 5e + 1,$$
$$4e = 5f + 1.$$

Эту систему легче всего решить, прибавив 4 к обеим частям каждого уравнения:

$$\begin{aligned} n + 4 &= 5(a + 1),\\ 4(a + 1) &= 5(b + 1),\\ 4(b + 1) &= 5(c + 1),\\ 4(c + 1) &= 5(d + 1),\\ 4(d + 1) &= 5(e + 1),\\ 4(e + 1) &= 5(f + 1). \end{aligned}$$

Перемножив соответственно левые и правые части всех уравнений, получим

$$4^5(n + 4)(a + 1)(b + 1)(c + 1)(d + 1)(e + 1) =$$
$$= 5^6(a + 1)(b + 1)(c + 1)(d + 1)(e + 1)(f + 1),$$

или

$$4^5(n + 4) = 5^6(f + 1).$$

Из последнего уравнения следует, что если оно имеет целое решение, то $n + 4$ и $f + 1$ должны делиться на 5^6 и 4^5 соответственно. Следовательно, $n + 4 = 5^6$ и $f + 1 = 4^5$ есть наименьшие натуральные числа, удовлетворяющие уравнению. Таким образом, наименьшее возможное n есть $5^6 - 4 = 15\,621$. (Очевидно, что все остальные неизвестные, т. е. a, b, c, d, e и f, также будут положительными целыми числами.)

Комментарии. Данное решение было предложено южноафриканским студентом Р. Гибсоном в 1958 году.

Как показано в первой части учебного раздела, некоторые головоломки можно решить, сведя их к математическим задачам. Решение, приведённое выше, — хороший пример такого подхода.

Различные версии этой головоломки известны уже очень давно. Аннотированная библиография Дэвида Сингмастера

[Sin10, Section 7.E] содержит больше дюжины страниц со ссылками на неё. Мартин Гарднер обсуждает её в своей колонке в *Scientific American*, а затем в [Gar87, часть 9]. В его тексте содержится несколько забавных анекдотов об истории головоломки, а также несколько методов её решения, включая остроумную идею — добавить четыре вымышленных или раскрашенных кокоса для упрощения вычислений.

103. Прыжки на другую сторону

Решение. Ответ — «нет».

Покрасим клетки доски в шахматном порядке (рис. 4.75).

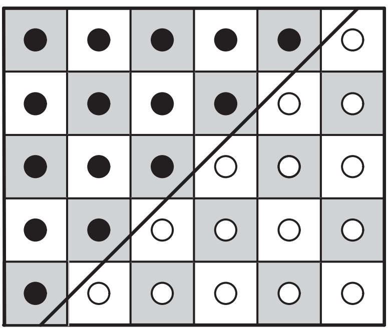


Рис. 4.75. Покрашенная доска для задачи «*Прыжки на другую сторону*»

Из 15 занятых шашками позиций девять — чёрные, в то время как из клеток, на которые нужно переместить шашки, лишь шесть чёрных. Поскольку цвет клетки, на которой стоит шашка, не меняется в результате любого из допустимых ходов, задача не имеет решения.

Комментарии. Задача основана на идее инварианта. Она является слегка изменённой версией задачи из книги [Gar97b, p. 335–336], предложенной Мартину Гарднеру читателем; Гарднер приписывает её авторство Марку Вегману из Исследовательского центра IBM им. Томаса Дж. Уотсона.

104. Разделение кучи фишек

Решение. а) Сумма произведений равна $\frac{(n-1)n}{2}$ вне зависимости от того, как происходит деление на кучки.

Если подсчитать сумму произведений для различных способов разделить кучу из n фишек на n кучек для нескольких первых натуральных n, можно заметить, что эта сумма не зависит от того, как делить на кучки. Что касается формулы для этой суммы произведений $P(n)$, то её можно получить двумя способами. Во-первых, рассматривая задачу для небольших n, можно заметить так называемые «треугольные числа» (суммы первых положительных целых чисел — см. вторую часть учебного раздела). Во-вторых, можно получить формулу, рассмотрев простейший способ деления на кучки: отделение одной фишки на каждом шаге. В последнем случае приходим к рекуррентному соотношению

$$P(n) = 1 \cdot (n-1) + P(n-1) \quad \text{при } n > 1, \quad P(1) = 0,$$

которое можно легко решить с помощью обратных подстановок:

$$P(n) = (n-1) + P(n-1) = (n-1) + (n-2) + P(n-2) = \ldots =$$
$$= (n-1) + (n-2) + \cdots + 1 + P(1) = \frac{(n-1)n}{2}.$$

Приведём прямое доказательство того факта, что $P(n)$ действительно равняется $\frac{n(n-1)}{2}$ и не зависит от способа разбиения на кучки, с помощью метода математической индукции. Если $n = 1$, то $P(1) = 0$. Предположим теперь, что утверждение верно для всех $1 \leqslant j < n$; мы докажем, что оно верно для $j = n$. Действительно, если начальную кучу из n фишек разбить на кучки из k и $n-k$ фишек, где $1 \leqslant k < n$, то по предположению индукции суммы для этих двух кучек равны соответственно $\frac{k(k-1)}{2}$ и $\frac{(n-k)(n-k-1)}{2}$. Следовательно, для суммы в нашем случае имеем выражение

$$P(n) = k(n-k) + \frac{k(k-1)}{2} + \frac{(n-k)(n-k-1)}{2} =$$
$$= \frac{2k(n-k) + k(k-1) + (n-k)(n-k-1)}{2} =$$
$$= \frac{2kn - 2k^2 + k^2 - k + n^2 - 2nk + k^2 - n + k}{2} =$$
$$= \frac{n^2 - n}{2} = \frac{n(n-1)}{2}.$$

б) Максимальная сумма сумм равна $\frac{n(n+1)}{2} - 1$.

Обозначив максимальную сумму, которую можно получить для исходной кучи из n фишек, через $M(n)$, получим следующее рекуррентное соотношение:

$$M(n) = n + \max_{1 \leqslant k \leqslant n-1} [M(k) + M(n-k)] \quad \text{при} \quad n > 1, \quad M(1) = 0. \tag{1}$$

Если найти значения $M(n)$ для первых нескольких n, естественно прийти к гипотезе, что сумма сумм максимальна при $k = 1$, и, следовательно,

$$M(n) = n + M(n-1) \quad \text{для} \quad n > 1, \qquad M(1) = 0.$$

Решение этого рекуррентного соотношения легко получить с помощью обратных подстановок:

$$M(n) = n + M(n-1) = n + (n-1) + M(n-2) = \ldots =$$
$$= n + (n-1) + \cdots + 2 + M(1) = \frac{n(n+1)}{2} - 1.$$

С помощью метода математической индукции несложно показать, что $M(n) = \frac{n(n+1)}{2} - 1$ действительно является решением рекуррентного соотношения (1). База индукции $M(1) = 0$ очевидна. Предположим теперь, что $M(j) = \frac{j(j+1)}{2} - 1$ удовлетворяет соотношению $M(j) = j + \max\limits_{1\leqslant k\leqslant j-1} [M(k) + M(j-k)]$ для всех $1 \leqslant j < n$. Покажем, что тогда $M(n) = \frac{n(n+1)}{2} - 1$ удовлетворяет соотношению $M(n) = n + \max\limits_{1\leqslant k\leqslant n-1} [M(k) + M(n-k)]$. Пользуясь предположением индукции, получаем

$$M(n) = n + \max_{1\leqslant k\leqslant n-1} [M(k) + M(n-k)] =$$
$$= n + \max_{1\leqslant k\leqslant n-1} \left[\frac{k(k+1)}{2} - 1 + \frac{(n-k)(n-k+1)}{2} - 1\right] =$$
$$= n + \max_{1\leqslant k\leqslant n-1} \left[k^2 - nk + \frac{n^2+n}{2} - 2\right].$$

Поскольку квадратичная функция $k^2 - nk + \frac{n^2+n}{2} - 2$ достигает своего минимума в $\frac{n}{2}$, середине промежутка $1 \leqslant k \leqslant n-1$, своих максимумов она достигает на концах промежутка — при $k = 1$ и $k = n-1$. Таким образом,

$$n + \max_{1\leqslant k\leqslant n-1} \left[k^2 - nk + \frac{n^2+n}{2} - 2\right] =$$
$$= n + \left[1^2 - n\cdot 1 + \frac{n^2+n}{2} - 2\right] =$$
$$= \frac{n(n+1)}{2} - 1,$$

что завершает доказательство.

Комментарии. Часть а) этой задачи см. в книге [Ros07, p. 292, Problem 14]; она основана на идее инварианта. Часть б) напоминает задачу динамического программирования.

105. Головоломка «MU»

Решение. С помощью допустимых преобразований из цепочки MI нельзя получить последовательность MU.

Если внимательно прочитать данные правила, легко понять, что любая строка, которую можно получить, начинается с символа M, и других M в строке нет. (Это наблюдение необходимо для пояснения правила 2.)

Теперь рассмотрим n — число символов I, которые могут быть в строке. Для начальной строки MI $n = 1$, n не делится на 3. Только правила 2 и 3 изменяют n — удваивают его и уменьшают на 3 соответственно. Ни одна из этих операций не может привести к числу, которое бы делилось на 3, если изначально оно на 3 не делилось. Так как для требуемой строки MU $n = 0$, т. е. делится на 3, такую строку нельзя получить из строки MI, для которой $n = 1$ не делится на 3.

Комментарии. Решение задачи основано на идее инварианта, упомянутой во второй части учебного раздела. Вопрос о том, какие строки можно получить, применяя некоторые правила, важен в информатике по нескольким причинам; в частности, высокоуровневые языки программирования определяются с использованием подобных правил.

Задача впервые появилась в первой части книги [Hof79] как пример формальной системы.

106. Лампочка и переключатели

Решение. Пронумеруем кнопки от 1 до n и воспользуемся следующим рекурсивным алгоритмом. Если $n = 1$ и свет выключен, нажмём кнопку 1. Если $n > 1$ и свет выключен, применим алгоритм рекурсивно к первым $n - 1$ кнопкам. Если в результате свет всё равно не включится, нажмём кнопку n и, если свет по-прежнему выключен, применим алгоритм к первым $n - 1$ кнопкам снова.

Рекуррентное соотношения для числа нажатий в худшем случае есть

$$M(n) = 2M(n-1) + 1 \quad \text{при } n > 1, \quad M(1) = 1,$$

а его решением является $M(n) = 2^n - 1$. (См. вторую часть учебного раздела, где решается такое же соотношение для задачи «*Ханойская башня*».)

Другой алгоритм решения состоит в следующем. Поскольку переключатель может находиться в одном из двух состояний, его можно считать битом в n-битовой строке, в которой 0 и 1 означают, скажем, начальное и противоположное начальному состояния переключателя соответственно. Общее число таких битовых строк (комбинаций состояний переключателей) равняется 2^n; одна из них соответствует начальному состоянию, а из остальных $2^n - 1$ битовых строк одна включает лампочку. В худшем случае придётся перебрать все эти $2^n - 1$ комбинаций переключателей. Чтобы добиться этого с помощью минимального числа нажатий на кнопки, каждое нажатие должно приводить к новой комбинации переключателей.

Есть несколько алгоритмов, начинающихся с битовой строки из n нулей и создающих все остальные $2^n - 1$ битовых строк, меняя по одному биту за раз. Самый известный из них — так называемый *рефлексивный двоичный код Грэя*, который можно построить следующим образом. Если $n = 1$, возвращаем список 0, 1. Если $n > 1$, рекурсивно создаём список битовых строк длины $n - 1$ и делаем копию этого списка; затем добавим 0 в начало каждой строки первого списка и 1 в начало каждой строки второго; наконец, присоединим второй список в обратном порядке к первому. Например, для $n = 4$ алгоритм выдаёт следующую последовательность битовых строк:

0000	0001	0011	0010	0110	0111	0101	0100
1100	1101	1111	1110	1010	1011	1001	1000

Возвращаясь к нашей задаче, мы можем пронумеровать переключатели от 1 до n справа налево и использовать последовательность битовых строк из кода Грэя для того, чтобы определить, какие кнопки нажимать: если следующая битовая строка отличается от предыдущей в i-м бите справа, то нажмём кнопку номер i. Например, в случае четырёх переключателей нам нужно будет нажимать кнопки в такой последовательности:

$$121312141213121.$$

Комментарии. Первое решение основано на стратегии «уменьшай на один». Второе использует сразу две стратегии: изменение представления (чтобы представить тумблеры и кнопки как битовые строки) и «уменьшай на один» (для создания кода Грэя).

Задача была предложена и решена Дж. Розенбаумом в 1938 году [Ros38] с использованием вышеизложенного метода — за годы до того, как был дан патент на изобретения кода Грэя. Мартин Гарднер упоминает эту задачу в своей статье о кодах Грэя [Gar86, p. 21]; в статью также включены их приложения к двум более известным головоломкам: «Китайские кольца» и «Ханойская башня». Кратко описав довольно запутанную историю кодов Грэя, Д. Кнут сделал вывод, что настоящим изобретателем этих кодов стоит считать француза Луи Гро, автора книги о «Китайских кольцах» 1872 года [Knu11, p. 284–285].

107. Лиса и заяц

Решение. Лиса может поймать зайца при чётных s и не может при нечётных s.

Рассмотрим чётность клеток $F(n)$ и $H(n)$, в которых, соответственно, лиса и заяц могут находиться перед n-м ходом. Для $n = 1$ имеем $F(1) = 1$, $H(1) = s$. На каждом шаге их позиция меняется — на 1 для лисы и на 3 для зайца. Следовательно, чётность их позиции меняется на противоположную на каждом шаге, а чётность разницы между их позициями остаётся неизменной. Таким образом, если начальная позиция зайца s нечётна, то разность между их позициями будет оставаться чётной перед каждым ходом лисы, и та никогда не окажется в соседней с зайцем позиции.

Не проиграет заяц и по причине отсутствия допустимых ходов, как могло бы произойти на доске короче 11 клеток в длину. Если $s = 3$, заяц может прыгнуть вправо и за два своих хода оказаться на позиции 9, и ситуация получится такой же, как либо при $s = 7$, если лиса за первые два хода оказывается на позиции 3, либо при $s = 9$, если лиса сначала двигается в позицию 2, а затем возвращается в позицию 1. Точно так же, если $s = 5$, заяц может первым ходом прыгнуть в позицию 8, и тогда ситуация вновь выходит такой же, как при $s = 7$. Пусть теперь $s = 7$, т. е. $F(1) = 1$ и $H(1) = 7$. После того как лиса первым ходом перемещается в позицию 2, заяц может прыгнуть в позицию 4. Если затем лиса возвращается в 1, заяц возвращается в 7; если лиса идёт в 3, заяц тоже прыгает обратно в 7 (не в 1, что привело бы к скорому поражению — лиса бы двинулась в 4, и у зайца не осталось бы допустимых ходов). После этого заяц может просто прыгать туда-обратно между 7 и 10, пока лиса не продвинется в 6, пытаясь его схватить. Затем заяц перескакивает через лису в 4 и начинает прыгать между 1 и 4. Если лиса возвращается в 5, заяц прыгает обратно в 7, и т. д. Таким образом, в этом случае лиса никогда не поймает зайца. Если s — нечётное целое число от 9 до 27 включительно, то та же стратегия, которая используется для $s = 7$, позволяет зайцу убежать от лисы. Если $s = 29$, то первый прыжок зайца в 26 сделает ситуацию аналогичной той, при которой лиса стартует из 1, а заяц — из 25.

Если s чётно, то лиса может вынудить зайца оказаться в позиции рядом с собой, просто постоянно двигаясь направо. Действительно, заяц стартует справа от лисы, а разность между их позициями нечётна. Если эта разность равна 1, то лиса просто ловит зайца следующим ходом. Если разность равна 3, лиса идёт направо, а у зайца остаётся выбор: либо прыгнуть по левую сторону от лисы и быть пойманным на следующем ходу, либо прыгнуть направо; если разность равняется 5 или больше, заяц может прыгнуть в обоих направлениях. Во всех этих случаях либо зайца можно поймать на следующем ходу, либо ситуация становится аналогичной «перезапуску» игры с укороченной на одну позицию доской, лисой в самой левой позиции и зайцем, находящимся на некоторой нечётной позиции справа от лисы. Следовательно, если лиса не поймает зайца прежде, чем достигнет позиции 26, она это сделает на следующем ходу.

Комментарии. Решение использует две идеи из области разработки и анализа алгоритмов: идею чётности и стратегию «уменьшай и властвуй».

Эта головоломка — модификация похожей игры из [Dyn71, с. 74, задача 54].

108. Самый длинный путь

Решение. Если считать, что расстояние между двумя соседними столбами равно 1, то самый длинный путь для любого $n \geqslant 2$ имеет длину

$$\frac{(n-1)n}{2} + \left\lfloor \frac{n}{2} \right\rfloor - 1.$$

Пронумеруем столбы последовательно от 1 до n. Легко видеть, что путь, получающийся применением жадной стратегии — т. е. путь $1, n, 2, n-1, \ldots, \left\lceil \frac{n}{2} \right\rceil$ для нечётных n и $1, n, 2, n-1, \ldots, \frac{n}{2}, \frac{n}{2}+1$ для чётных n — можно сделать длиннее, заменив последний перегон более длинным, соединяющим последний столб «жадного пути» со столбом 1 (см. рис. 4.76, иллюстрирующий случаи $n = 5$ и $n = 6$). Доказательство того, что улучшенный таким образом «жадный путь» действительно является самым длинным, не сложное, но довольно громоздкое; его можно найти у Гуго Штейнгауза [Ste64].

Самый длинный путь для $n = 5$ есть $3 \to 1 \to 5 \to 2 \to 4$; для $n = 6$ самым длинным путём является $4 \to 1 \to 6 \to \to 2 \to 5 \to 3$. Решение задачи не единственно для $n > 4$, но любой самый длинный путь должен либо начинаться, либо заканчиваться в одном из трёх серединных столбов при нечётных n и в одном из двух серединных столбов при чётных n.

Комментарии. Эта головоломка — интересный пример того, как жадный алгоритм даёт неоптимальное, но легко улучшающееся до оптимального решение.

Как сказано выше, задача взята из книги «Сто задач» Гуго Штейнгауза [Ste64, задача 64]. Она также была включена в одну из статей Мартина Гарднера в *Scientific American*, которая впоследствии была перепечатана в [Gar71, p. 235, 237–238].

109. Домино «дубль-*n*»

Решение. а) Набор домино «дубль-n» состоит из $n+1$ костяшек вида $(0, j)$, где $0 \leqslant j \leqslant n$, n костяшек вида $(1, j)$, где $1 \leqslant j \leqslant n, \ldots$, и, наконец, одной костяшки (n, n). Таким образом, общее число костяшек равняется $(n+1) + n + \ldots + 1 = \frac{(n+1)(n+2)}{2}$.

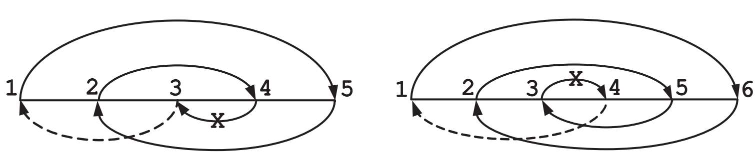


Рис. 4.76. Самые длинные пути, получающиеся при улучшении «жадного решения», для $n = 5$ и $n = 6$

б) Для любого значения k, $0 \leqslant k \leqslant n$, есть n костяшек, одна половина которых содержит k точек, а другая — отличное от k число точек, а также одна костяшка, на обеих половинах которой по k точек. Поэтому общее число половинок с k точками равняется $n+2$. Следовательно, общее число точек на всех костяшках равно

$$\sum_{k=0}^{n} k(n+2) = \frac{n(n+1)(n+2)}{2}.$$

в) Требуемое кольцо можно построить в том и только том случае, когда n — положительное чётное число. Поскольку каждая пара соседних костяшек в кольце должна иметь одинаковое количество точек на прилегающих половинках, число костяшек с половинами, на которых k точек, должно быть чётным для всех $0 \leqslant k \leqslant n$. Поскольку это число равняется $n+2$ для любого k (см. часть б)), такое кольцо не может существовать при нечётном n.

Когда n чётно и положительно, требуемое кольцо можно построить следующим образом. Если $n = 2$, то это кольцо

$$R(2)\colon\ (0,0)(0,1)(1,1)(1,2)(2,2)(2,0).$$

Если $n = 2s$, где $s > 1$, рекурсивно построим кольцо $R(2s-2)$ из всех домино из поднабора «дубль-$(2s-2)$». Затем построим цепочку, включающую в себя оставшиеся домино вида (i,j), где $j = 2s-1$ и $2s$, $0 \leqslant i \leqslant j$, например из последовательных четвёрок $(2t, 2s-1)(2s-1, 2t+1)(2t+1, 2s)(2s, 2t+2)$ для $t = 0, 1, \ldots, s-1$, после которых расположена $(2s, 0)$. Наконец, поместим эту цепочку между соседними $(0,0)$ и $(0,1)$ в $R(2s-2)$ и получим кольцо, состоящее из всех домино «дубль-$2s$».

Другой подход к решению — свести задачу к вопросу о существовании эйлерова цикла в полном графе с $n+1$ вершинами. Вершина i, $0 \leqslant i \leqslant n$, в таком графе означает возможное число точек на одной из двух половинок домино, а ребро между вершинами i и j означает домино с i и j на половинках. Домино с одинаковым числом точек на половинах можно либо не рассматривать до тех пор, пока не будет построено кольцо, включающее в себя все остальные домино, а затем вставить между любыми двумя подходящими домино, либо представить их как петли (рёбра, соединяющие некоторую вершину графа саму с собой). Очевидно, что эйлеров цикл в таком графе будет обозначать кольцо из всех домино дубль-n, верно и обратное. Согласно хорошо известной теореме — см., например, вторую часть учебного раздела — в связном графе эйлеров цикл существует тогда и только тогда, когда у всех его вершин чётные степени. То есть для нашего графа тогда и только тогда, когда n чётно. Алгоритм построения эйлерова цикла вкратце изложен в решении задачи «*Обвести фигуру*» (№ 28).

Комментарии. Решение головоломки использует несколько стратегий: чётность, «уменьшай и властвуй», а во втором способе решения применяется сведение задачи к более простой. Роуз Болл [Bal87, p. 251] приписывает последнюю идею французскому математику Гастону Тарри, который использовал её, чтобы найти число возможных перестановок набора домино «дубль-n».

110. Хамелеоны

Решение. Ответ — «нет».

После встречи двух хамелеонов разного цвета число хамелеонов их цветов уменьшится на 1, а число хамелеонов третьего цвета — увеличится на 2. Следовательно, одна из разностей между числом по-разному окрашенных хамелеонов не изменится, а две другие — увеличатся на 3. Таким образом, остаток от деления на 3 всех этих разностей не изменится. Отсюда сразу следует, что для данных из условия задачи невозможна ситуация, в которой все хамелеоны были бы одного цвета. Действительно, начальные разности равны 4, 1 и 5, и поэтому их остатки от деления на 3 суть 1, 1 и 2. Но если бы все хамелеоны могли стать одного цвета, одна из разностей должна была бы быть нулевой, как и её остаток от деления на 3.

Комментарии. Эта головоломка основана на применении идеи инварианта в довольно-таки необычном ключе. Разные версии этой задачи были включены в несколько сборников головоломок (например, [Hes09, Problem 24] и [Fom96, с. 130, задача 21]).

СЛОЖНЫЕ ГОЛОВОЛОМКИ

111. Переворачивание треугольника из монеток

Решение. Минимальное число перемещений монет, позволяющее перевернуть треугольник из $T_n = \frac{n(n+1)}{2}$ монет, равно $\left\lfloor \frac{T_n}{3} \right\rfloor$. Чтобы перевернуть треугольник за минимальное число ходов, нам, очевидно, нужно использовать один из его горизонтальных рядов как основание для перевёрнутого треугольника. Рассмотрим k-й горизонтальный ряд данного треугольника, $1 \leqslant k \leqslant n$. (Мы считаем, что ряды пронумерованы сверху вниз, так что в k-м ряду k монет.) Найдём минимальное число ходов, которое необходимо, чтобы перевернуть треугольник и сделать этот ряд основанием перевёрнутого треугольника. Чтобы это сделать, необходимо переместить в этот ряд $n-k$ монет. Разумно взять эти монеты из последнего ряда следующим образом: $\left\lceil \frac{n-k}{2} \right\rceil$ самых левых монет и $\left\lfloor \frac{n-k}{2} \right\rfloor$ самых правых монет из последнего ряда переместить в правый и левый края ряда k соответственно; тогда в последнем ряду останется ровно $n - \left(\left\lceil \frac{n-k}{2} \right\rceil + \left\lfloor \frac{n-k}{2} \right\rfloor\right) = n - \left(n-k\right) = k$ монет. Затем нужно переместить $n-k-2$ монет в ряд $k+1$ данного треугольника; их разумно взять из предпоследнего ряда: его $\left\lceil \frac{n-k}{2} \right\rceil - 1$ самых левых монет и $\left\lfloor \frac{n-k}{2} \right\rfloor - 1$ самых правых монет перемещаем в правый и левый края ряда $k+1$ соответственно. Повторяем эту операцию до тех пор, пока не передвинем $\left\lceil \frac{n-k}{2} \right\rceil - \left\lfloor \frac{n-k}{2} \right\rfloor$ (что равно 0 или 1 в зависимости от того, чётно $n-k$ или нет) самых левых монет из ряда $n - \left\lfloor \frac{n-k}{2} \right\rfloor$ в правый конец ряда $k + \left\lfloor \frac{n-k}{2} \right\rfloor$. Наконец, монеты из первых $k-1$ рядов данного треугольника следует переместить в обратном порядке, т. е. от самого длинного к самому короткому, чтобы образовать последовательно все ряды под рядом, изначально служившим основанием.

На рис. 4.77 изображено решение для случая чётного $n-k$. Если $n-k$ нечётно, можно также сдвинуть из правого

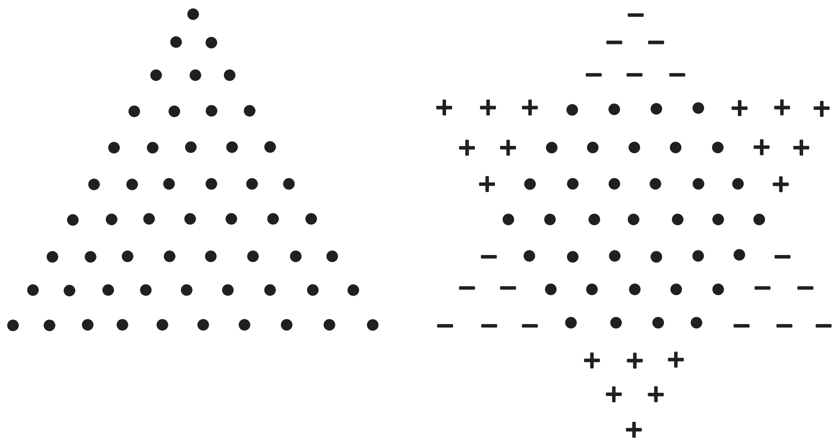


Рис. 4.77. Переворачиваем треугольник из T_{10} монет, делая ряд $k = 4$ основанием перевёрнутого треугольника (+, – и кружки обозначают, соответственно, центры добавленных, перемещённых и остающихся на месте монет)

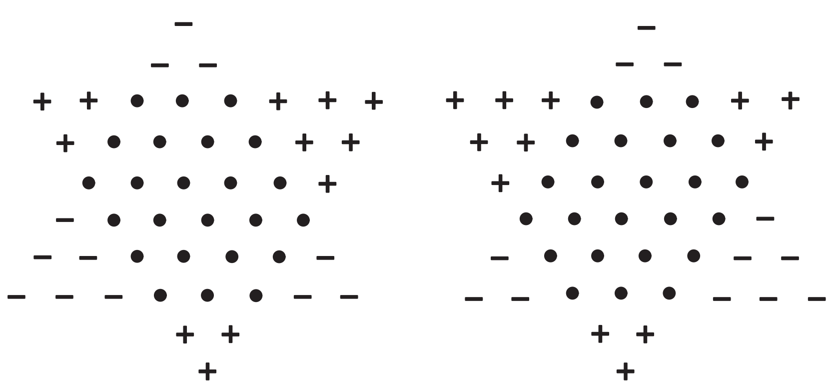


Рис. 4.78. Переворачиваем треугольник из T_8 монет, делая ряд $k = 3$ основанием перевёрнутого треугольника

конца каждого ряда на одну монету больше, чем из левого конца, и получить решение, симметричное тому, что описано выше. Эти два решения иллюстрируются рис. 4.78.

Общее число шагов $M(k)$, которые делает этот алгоритм, очевидно является минимальным необходимым для того, чтобы сделать k-й ряд основанием перевёрнутого треугольника, поскольку каждое перемещение монеты увеличивает число монет в ряду, который надо удлинить, и одновременно уменьшает число монет в ряду, который надо укоротить.

Значение $M(k)$ можно вычислить следующим образом:

$$M(k)=\sum_{j=0}^{\lfloor\frac{n-k}{2}\rfloor}(n-k-2j)+\sum_{j=1}^{k-1}j=\sum_{j=0}^{\lfloor\frac{n-k}{2}\rfloor}(n-k)-\sum_{j=0}^{\lfloor\frac{n-k}{2}\rfloor}2j+\sum_{j=1}^{k-1}j=$$
$$=(n-k)\left(\left\lfloor\frac{n-k}{2}\right\rfloor+1\right)-\left\lfloor\frac{n-k}{2}\right\rfloor\left(\left\lfloor\frac{n-k}{2}\right\rfloor+1\right)+\frac{(k-1)k}{2}=$$
$$=\left(\left\lfloor\frac{n-k}{2}\right\rfloor+1\right)\left\lceil\frac{n-k}{2}\right\rceil+\frac{(k-1)k}{2}.$$

Если $n-k$ чётно, формулу для $M(k)$ можно дополнительно упростить:

$$M(k)=\left(\frac{n-k}{2}+1\right)\frac{n-k}{2}+\frac{(k-1)k}{2}=$$
$$=\frac{3k^2-(2n+4)k+n^2+2n}{4}.$$

Если $n-k$ нечётно, формула для $M(k)$ принимает вид

$$M(k)=\left(\frac{n-k-1}{2}+1\right)\frac{n-k+1}{2}+\frac{(k-1)k}{2}=$$
$$=\frac{3k^2-(2n+4)k+(n+1)^2}{4}.$$

В обоих случаях минимум квадратичной функции $M(k)$ достигается при $k=\frac{n+2}{3}$. Поэтому, если $\frac{n+2}{3}$ есть целое число (т. е. если $n=3i+1$), то у задачи существует единственное решение (с точностью до монет, которые остаются на месте), потому что тогда $n-k=(3i+1)-\frac{3i+1+2}{3}=2i$ чётно (см. рис. 4.77 для примера).

Если $\frac{n+2}{3}$ не целое, то у задачи есть два качественно различных решения[1)], которые получаются применением

[1)]Мы считаем решения качественно различными, если они используют в качестве основания разные ряды начального треугольника. Одно из таких решений — то, при котором из основания данного треугольника в основание перевёрнутого передвигается нечётное число монет — будет иметь симметричный аналог (для примера см. рис. 4.78).

вышеописанного алгоритма к $\frac{n+2}{3}$, округлённому вверх или вниз, т. е. при $k^+ = \left\lceil \frac{n+2}{3} \right\rceil$ и $k^- = \left\lfloor \frac{n+2}{3} \right\rfloor$.

Минимальное число перемещений монет также можно выразить формулой, приведённой в [Gar89, p. 23], [Tri69] и [Epe70]:

$$\left\lfloor \frac{n(n+1)}{6} \right\rfloor = \left\lfloor \frac{T_n}{3} \right\rfloor,$$

где $T_n = \frac{n(n+1)}{2}$ есть общее число монет в треугольнике. Это легко проверить, подставив все три возможных значения n ($n = 3i$, $n = 3i + 1$ и $n = 3i + 2$) в формулу и убедившись, что она действительно приводит к тем же ответам, что были получены выше для оптимально выбранного k.

Комментарии. Как подчёркивается в первой части учебного раздела, сведение головоломки к математической задаче есть одна из разновидностей стратегии разработки алгоритмов «преобразуй и властвуй».

Этой головоломке уже много лет. Например, Мэкси Брук включил случай 10 монет в свою книгу [Bro63], причём ссылается он на неё как на способ «выиграть кофе» — пари о том, что собеседник не сможет найти решение из трёх ходов, вы скорее выиграете, чем проиграете (p. 15).

Мартин Гарднер попросил читателей своей мартовской колонки в *Scientific American* 1966 года найти компактную формулу для минимального числа монет, которые нужно передвинуть для решения этой задачи в общем случае. В качестве решения он использовал следующий «геометрический» подход ([Gar89, p. 23]): «Работая над задачей в общем виде, рассматривая равносторонние конструкции произвольного размера, читатель, возможно, заметил, что задача заключается в том, чтобы нарисовать ограничивающий треугольник (подобный рамке, в которой находятся 15 шаров перед началом игры в бильярд), перевернуть его и накрыть им фигуру так, чтобы как можно большее число монет оказалось внутри него. В любом случае наименьшее число монет, которые нужно передвинуть, чтобы перевернуть фигуру, получается, если разделить число монет на 3 и отбросить остаток». И Тригг [Tri69], и Эперсон [Epe70] описывают решение в терминах треугольников, которые нужно отрезать от данного треугольника, но ни один из авторов не доказывает оптимальность такого решения. Используя отрезаемые треугольники, легко модифицировать алгоритм, обрисованный выше, чтобы выполнялось дополнительное условие: «на каждом ходу монета должна быть перемещена так, чтобы коснуться двух монет, по которым можно однозначно определить её новое положение» [Gar89, с. 13]. Вместо того чтобы двигать монеты ряд за рядом, можно передвинуть все монеты в отрезанном треугольнике к позициям, которые они должны занять в перевёрнутом треугольнике: достаточно всегда брать внешнюю монету из данного треугольника и перемещать её в такое место, чтобы она касалась хотя бы двух других монет. Для общего исследования условий, при которых одна фигура из монет может быть преобразована в другую при таком ограничении, см. [Dem02].

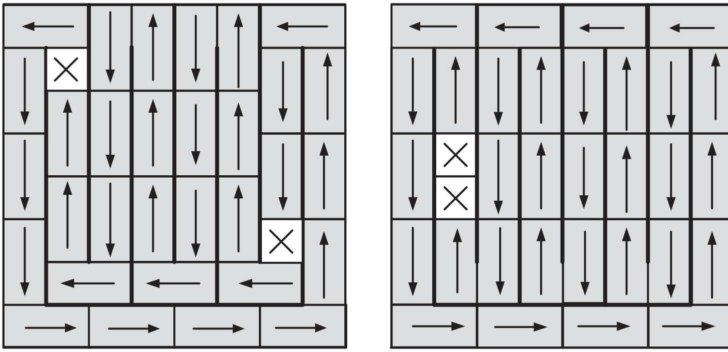
112. Снова о покрытии домино

Решение. Головоломка имеет решение тогда и только тогда, когда n чётно.

Разумеется, у задачи нет решения при нечётных n из соображений чётности: тогда число квадратов, которое требуется покрыть, нечётно, в то время как домино могут покрыть только чётное число квадратов.

Если n чётно, то шахматную доску $n \times n$ без двух произвольных квадратов $1{\times}1$ разных цветов всегда можно покрыть домино. Для $n = 2$ это, очевидно, верно. При $n > 2$ для доказательства потребуется хитроумная конструкция, которую называют *перегородками Гомори* (*Gomory barriers*) в честь её изобретателя, американского математика Ральфа Гомори. Эти перегородки разбивают доску на два «коридора» с концами в удалённых полях, если они не соседние, или лишь на один «коридор» в противном случае (см. рис. 4.79, *а* и *б*). Покрывающие домино всегда можно положить, причём единственным способом, вдоль этих «коридоров».

Комментарии. Перегородки Гомори в этом решении строятся не единственным образом. Соломон Голомб приводит в своей книге о полимино четыре схемы, отмечая, что «есть ещё буквально сотни других» [Gol94, p. 112]. Конечно, если два удалённых квадрата на доске $n \times n$, где n чётно, имеют один цвет — например, это два противоположных угла



(*а*) (*б*)

Рис. 4.79. Покрытие костяшками домино доски 8×8 с двумя удалёнными квадратами с использованием перегородок Гомори. (*а*) Два удалённых поля не соседние. (*б*) Два удалённых поля находятся рядом

на диагонали (головоломка «*Покрытие фигурками домино шахматных досок с дефектами*» во второй части учебного раздела) — то замощение невозможно, поскольку число чёрных и белых квадратов, которые необходимо покрыть, различно.

113. Исчезающие монеты

Решение. Головоломку можно решить в том и только том случае, когда число монет Н, лежащих орлами вверх, в начальной линии нечётно. Если это так, задачу можно решить, раз за разом убирая самого левого орла до тех пор, пока не останется ни одной монеты.

Докажем, что такой алгоритм действительно приводит к решению для линии из n монет с единственным орлом. Если орёл находится на одном из концов линии (например, левом), то после его удаления получится такая же линия, но уже из $n-1$ монет:

$$\underbrace{\mathbf{H}\ \ \mathrm{T}\ \ \ldots\ \ \mathrm{T}}_{n} \quad \Longrightarrow \quad \underbrace{\mathbf{H}\ \ \mathrm{T}\ \ \ldots\ \ \mathrm{T}}_{n-1}$$

Следовательно, если повторить это действие n раз, мы уберём все монеты.

Если единственный орёл находится не на конце линии, то после его удаления получатся две более короткие линии, на одном из концов каждой будет единственный в линии орёл, с пробелом между ними:

$$\underbrace{\mathrm{T} \ldots \mathrm{T}\ \mathrm{T}\ \mathbf{H}\ \mathrm{T}\ \mathrm{T} \ldots \mathrm{T}}_{n} \implies \underbrace{\mathrm{T} \ldots \mathrm{T}\ \mathrm{H}\ \_\ \mathrm{H}\ \mathrm{T} \ldots \mathrm{T}}_{n-1}$$

Затем можно решить задачу, удалив все монеты из этих коротких линий способом, показанным выше.

Рассмотрим теперь общий случай, когда мы убираем самого левого орла из линии с нечётным числом орлов, большим 1. Пусть $k \geqslant 0$ — число решек, находящихся слева от самого левого орла. Вне зависимости от того, идёт ли справа от этого орла решка или другой орёл, после его удаления получится одна линия с $k-1$ решками и затем одним орлом (эта линия либо пустая (при $k=0$), либо может быть удалена вышеописанным способом) и более короткая линия с нечётным числом орлов, которую можно убрать аналогично:

$$\underbrace{\mathrm{T} \ldots \mathrm{T}\ \mathrm{T}}_{k}\quad \underbrace{\mathbf{H}\ \mathrm{T}\ \ldots}_{\text{нечётное кол-во H}} \implies \underbrace{\mathrm{T} \ldots \mathrm{T}}_{k-1}\ \mathrm{H}\ \_\quad \underbrace{\mathrm{H}\ \ldots}_{\text{нечётное кол-во H}}$$

$$\underbrace{\mathrm{T} \ldots \mathrm{T}\ \mathrm{T}}_{k}\quad \underbrace{\mathbf{H}\ \mathrm{H}\ \ldots}_{\text{нечётное кол-во H}} \implies \underbrace{\mathrm{T} \ldots \mathrm{T}}_{k-1}\ \mathrm{H}\ \_\quad \underbrace{\mathrm{T}\ \ldots}_{\text{нечётное кол-во H}}$$

Вторая часть доказательства — того факта, что для линии с чётным числом орлов у головоломки нет решения — аналогична первой. Если число орлов нулевое, то задачу нельзя решить просто потому, что можно удалять только орлов. Если в линии есть орлы, то удаление одного из них либо приведёт к одной более короткой линии с чётным числом орлов, либо к двум коротким линиям и пробелу между ними, причём хотя бы в одной из линий будет чётное число орлов.

Комментарии. В то время как основная идея, очевидно, базируется на стратегии «уменьшай на один», в известном смысле неожиданная помощь приходит со стороны стратегии «разделяй и властвуй». Также обратите внимание, что удаление произвольного орла может не привести к решению: например, если удалить среднего орла из линии из трёх орлов, получатся две решки, которые удалить невозможно.

Головоломка появилась в книге [Tan01, Problem 29.4] со ссылкой на её цикличный вариант, предложенный

Д. Беквитом [Bec97]. Также задача была включена в книгу [Ste09, p. 245].

114. Обход точек

Решение. На рис. 4.80 представлены решения для $n = 3$, 4 и 5. Они показывают, что путь из $2n - 2$ сегментов через n^2 точек можно получить из пути, состоящего из $2(n - 1) - 2$ сегментов и проходящего через $(n - 1)^2$ точек, добавив к последнему два сегмента — горизонтальный и вертикальный — проходящих через следующие столбец и строку из n точек.

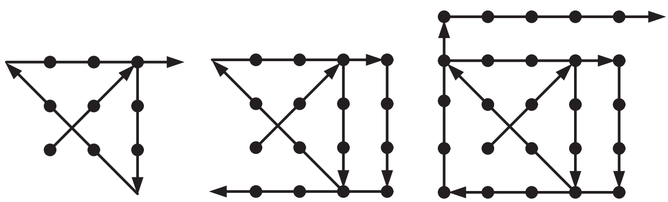


Рис. 4.80. Путь из $2n - 2$ сегментов через n^2 точек для $n = 3$, $n = 4$ и $n = 5$

Комментарии. После того, как решение найдено для случая $n = 3$, алгоритм построения пути использует стратегию «уменьшай и властвуй» наоборот. Случай $n = 3$ — одна из головоломок, любимых во все времена. Задача, в силу структуры её решения, считается источником выражения «thinking outside the box» (что означает «нестандартное мышление», «выход за рамки задачи»). И Генри Дьюдени, и Сэм Лойд публиковали её около века назад. Дьюдени также рассматривал случаи $n = 7$ и $n = 8$ [Dud58, задачи 329–332]. Приведённый здесь общий случай включён в «Задачи с изюминкой» Чарльза Тригга [Tri85, задача 261], со ссылкой на решение М. С. Кламкина, опубликованное в феврале 1955 года в *American Mathematical Monthly* (p. 124).

115. Задача Баше о гирях

Решение. Ответом являются n последовательных степеней двойки, начиная с 2, и n последовательных степеней тройки, начиная с 3, для частей а) и б) соответственно.

а) Применим жадный подход к случаям первых нескольких n. При $n = 1$ нужно использовать $w_1 = 1$, чтобы взвесить груз массой 1. При $n = 2$ мы просто добавляем $w_2 = 2$, чтобы можно было взвесить груз массой 2. С помощью набора гирь $\{1,2\}$ можно достичь баланса чаш весов для каждого целого веса вплоть до их суммы 3. Для $n = 3$, в духе жадного мышления, мы берём следующий ранее недостижимый вес: $w_3 = 4$. С помощью трёх гирь $\{1,2,4\}$ можно взвесить груз любой целочисленной массы l от 1 до 7, причём двоичное представление l показывает те гири, которые нужно взять, чтобы взвесить l:

груз массой l	1	2	3	4	5	6	7
l в двоичной записи	1	10	11	100	101	110	111
гири для взвешивания l	1	2	$2+1$	4	$4+1$	$4+2$	$4+2+1$

Обобщая эти рассуждения, мы можем предположить, что для любого положительного целого n с помощью набора гирь с массами — последовательными степенями двойки $\{w_i = 2^i, i = 0,1,\ldots,n-1\}$ можно взвесить любой целый груз в наибольшем возможном промежутке от 1 до их суммы $\sum_{i=0}^{n-1} 2^i = 2^n - 1$ включительно. Тот факт, что каждый целый вес l в промежутке $1 \leqslant l \leqslant 2^n - 1$ можно взвесить с помощью этого набора, немедленно следует из двоичного представления l, в котором видны те гири, что нужны для уравновешивания l. Так как в любом наборе из n гирь всего $2^n - 1$ различных подмножеств (меньше, если некоторые гири имеют одинаковый вес), которые можно положить на одну чашу весов, то с их помощью нельзя взвесить более $2^n - 1$ разных грузов. Это доказывает, что никакой набор из n гирь не мог бы работать для большего числа последовательных целых значений массы груза, чем $1 \leqslant l \leqslant 2^n - 1$, если гири можно класть только на свободную чашу весов.

б) Если гири разрешается класть на обе чаши весов, то с помощью n гирь для $n > 1$ можно достичь большего диапазона доступных для взвешивания грузов. Для $n = 1$, конечно, единственный вариант — гиря массой 1.

Набор гирь с массами $\{1,3\}$ позволит взвесить любой целый груз вплоть до 4. Гири с массами $\{1,3,9\}$ позволят взвесить любой целый груз вплоть до 13, как показывает следующая таблица:

груз массой l	1	2	3	4	5
l в троичной записи	1	2	10	11	12
гири для взвешивания l	1	$3-1$	3	$3+1$	$9-3-1$

груз массой l	6	7	8	9
l в троичной записи	20	21	22	100
гири для взвешивания l	$9-3$	$9+1-3$	$9-1$	9

груз массой l	10	11	12	13
l в троичной записи	101	102	110	111
гири для взвешивания l	$9+1$	$9+3-1$	$9+3$	$9+3+1$

В общем случае набор гирь $\{w_i = 3^i, i = 0, 1, \ldots, n-1\}$ позволит взвесить любой целый груз от 1 до их суммы $\sum_{i=0}^{n-1} 3^i = \frac{3^n - 1}{2}$ включительно. Представление массы груза в троичной системе показывает, какие гири нужно использовать, следующим образом. Если троичная запись числа l, $l \leqslant \frac{3^n-1}{2}$, содержит только нули и единицы, то для взвешивания груза следует положить гири, соответствующие единицам, на противоположную чашу весов. Если троичная запись числа l содержит одну или более двоек, то можно заменить каждую двойку на $(3-1)$ и записать l единственным образом в «сбалансированной троичной системе» (см. [Knu98, p. 207–208]):

$$l = \sum_{i=0}^{n-1} \beta_i 3^i, \text{ где } \beta_i \in \{0, 1, -1\}.$$

Например,

$$5 = 12_3 = 1 \cdot 3^1 + 2 \cdot 3^0 = 1 \cdot 3^1 + (3-1) \cdot 3^0 = 2 \cdot 3^1 - 1 \cdot 3^0 =$$
$$= (3-1) \cdot 3^1 - 1 \cdot 3^0 = 1 \cdot 3^2 - 1 \cdot 3^1 - 1 \cdot 3^0.$$

(Заметим, что если мы начинаем с самой правой двойки, то после упрощения новая самая правая двойка, если будет присутствовать в разложении, будет находиться на некоторой позиции слева от начальной. Это доказывает, что после конечного числа таких замен мы сможем убрать все двойки.)

Используя представление $l = \sum_{i=0}^{n-1} \beta_i 3^i$, где $\beta_i \in \{0, 1, -1\}$, мы можем взвесить груз l, поместив все гири массой $w_i = 3^i$ для отрицательных β_i вместе с грузом на одну чашу весов, а все гири массой $w_i = 3^i$ для положительных β_i — на другую чашу. Наконец, любую гирю из набора n гирь можно поместить либо на левую чашу, либо на правую, либо не помещать ни на одну из них. Следовательно, есть $3^n - 1$ способов использовать n гирь для взвешивания груза целочисленной массы. Принимая во внимание симметрию, получаем, что с их помощью можно взвесить не более $\dfrac{3^n-1}{2}$ различных грузов. Это доказывает, что никакой набор из n гирь не мог бы работать для большего числа последовательных целых значений массы груза, чем $1 \leqslant l \leqslant \dfrac{3^n-1}{2}$.

Комментарии. Эта головоломка даёт хороший пример применимости жадного подхода, а также показывает, что порой полезно использовать недесятичные системы счисления. Также следует отметить, что если не требуется точно сбалансировать каждый целый груз, то с помощью того же числа гирь можно взвесить вдвое большее число грузов. Например, с помощью четырёх гирь 2, 6, 18 и 54 можно взвесить любой целый груз от 1 до 80, причём грузы с чётными массами 2, 4, …, 80 можно точно сбалансировать, в то время как любой груз с нечётной массой можно взвесить, найдя две последовательные комбинации гирь 2, 6, 18 и 54, одна из которых легче, а другая тяжелее рассматриваемого груза. Например, если целый груз тяжелее 10, но легче 12, то он, очевидно, весит 11 [Sin10, Section 7.L.3, p. 95].

Головоломка названа в честь Клода Гаспара Баше де Мезириака, автора книги «Problèmes» [Bac12] — книги-пионера в области занимательной математики, опубликованной в 1612 году и содержащей решение этой задачи. Современные исследователи истории занимательной математики считают Хасиба Табари (ок. 1075) и Фибоначчи (1202 г.) первыми математиками, которые решили эту задачу [Sin10, Sections 7.L.2.c, 7.L.3].

116. Подсчёт «пустых номеров»

Решение. Ответом к части а) является $2^{\lceil \log_2 n \rceil} - n$; ответом к части б) — количество единиц в двоичном представлении числа $2^{\lceil \log_2 n \rceil} - n$.

а) Если мы переносим минимальное число игроков сразу во второй раунд, так чтобы в этом раунде осталось число игроков, равное степени двойки, то число пустых номеров равняется $2^{\lceil \log_2 n \rceil} - n$. Например, для $n = 10$ число пустых номеров равно $2^{\lceil \log_2 10 \rceil} - 10 = 6$.

Действительно, пусть $2^{k-1} < n \leqslant 2^k$, где n — число игроков. Если b игроков получают пустой номер, то $n - b$ игроков участвуют в первом раунде и $\frac{n-b}{2}$ победителей проходят во второй раунд. Таким образом, имеем уравнение $b + \frac{n-b}{2} = 2^{k-1}$. Его решением является $b = 2^k - n$, где $k = \lceil \log_2 n \rceil$. (Заметим, что $n - b = 2n - 2^k$ и, следовательно, чётно, как и следовало ожидать.)

б) Обозначим через n число игроков — участников турнира ($2^{k-1} < n \leqslant 2^k$), а через $B(n)$ — общее число пустых номеров, если мы переносим минимальное число игроков в каждом раунде, так чтобы в каждом раунде было чётное число игроков. Другими словами, если число игроков чётно, в этом раунде нет пустых номеров; если же оно нечётно, то один из игроков получает пустой номер и в следующем раунде оказывается $1 + \frac{n-1}{2} = \frac{n+1}{2}$ игроков. Таким образом, имеем следующее рекуррентное соотношение для $B(n)$:

$$B(n) = \begin{cases} B\left(\frac{n}{2}\right), & \text{если } n > 0 \text{ чётное,} \\ 1 + B\left(\frac{n+1}{2}\right), & \text{если } n > 1 \text{ нечётное,} \end{cases} \qquad B(1) = 0.$$

Хотя для решения этого соотношения нет явной формулы, его можно получить с помощью следующего алгоритма, упомянутого Мартином Гарднером в книге «Есть идея!» [Gar78, p. 6]: $B(n)$ можно вычислить, подсчитав количество единиц в двоичном представлении разности $b(n) = 2^k - n$, где $k = \lceil \log_2 n \rceil$. Другими словами, число пустых номеров в условиях части б) равняется числу единиц в двоичном представлении числа пустых номеров, которое получается в части а). Например, для $n = 10$ при применении этого алгоритма вычисляем $b(10) = 2^4 - 10 = 6 = 110_2$ и, таким образом, пустых номеров будет 2, в то время как для $n = 9$ после подсчёта $b(9) = 2^4 - 9 = 7 = 111_2$ получаем в ответе 3 пустых номера.

Пусть $G(n)$ есть число, получающееся при применении алгоритма Гарднера для положительного целого n $(2^{k-1} < n \leqslant 2^k)$; мы покажем, что $G(n)$ удовлетворяет рекуррентному соотношению и начальному условию, приведённым выше для числа пустых номеров $B(n)$.

Если $n = 1$, то $2^0 - 1 = 0$ и, следовательно, $G(1)$, число единиц в двоичном представлении нуля, равно 0.

Пусть n — положительное чётное число; $2^{k-1} < n \leqslant 2^k$ и, следовательно, $2^{k-2} < \frac{n}{2} \leqslant 2^{k-1}$. Поскольку $b(n) = 2^k - n$ в этом случае чётно, самая правая цифра в его двоичном представлении есть 0. Таким образом, $b\left(\frac{n}{2}\right) = 2^{k-1} - \frac{n}{2} = \frac{b(n)}{2}$ имеет такое же число единиц, как $b(n)$. Следовательно,

$$b\left(\frac{n}{2}\right) = 2^{k-1} - \frac{n}{2} = \frac{b(n)}{2}$$

для любого положительного чётного n

Пусть n — нечётное целое число, большее 1; $2^{k-1} < n \leqslant 2^k$ и, следовательно, $2^{k-2} < \frac{n+1}{2} \leqslant 2^{k-1}$. По определению $G\left(\frac{n+1}{2}\right)$ равняется числу единиц в двоичном представлении $b\left(\frac{n+1}{2}\right) = 2^{k-1} - \frac{n+1}{2}$. Поскольку $b(n) = 2^k - n$ в этом случае нечётно, самая правая цифра в его двоичном представлении есть 1. Если отбросить эту последнюю единицу в двоичном представлении числа $2^k - n$, получим двоичное представление числа

$$\frac{2^k - n - 1}{2} = 2^{k-1} - \frac{n+1}{2} = b\left(\frac{n+1}{2}\right).$$

Следовательно, для любого нечётного $n > 1$ получаем $G(n) = 1 + G\left(\frac{n+1}{2}\right)$ — именно это и требовалось показать.

Комментарии. Наше решение части б) основано на интерпретации турнира на выбывание как алгоритма «уменьшай вдвое». Можно получить другое решение, добавив $2^{\lceil \log_2 n \rceil} - n$ воображаемых игроков, чтобы получить более простой частный случай. Тогда число пустых номеров получается, если подсчитать число игр между воображаемыми и настоящими игроками. Этот подход приводит к альтернативному выражению для числа пустых номеров — оно равняется количеству нулей в двоичном представлении числа $n-1$, где $n>1$ — число настоящих игроков в турнире (см. [MathCentral], задача месяца в октябре 2009 года).

Между прочим, турнирные деревья, помимо того что удобны для организации турниров, имеют приложения в информатике (см., например, [Knu98]).

117. Одномерный солитёр

Решение. Если клетки доски пронумерованы последовательно от 1 до n, то изначально пустовать может клетка 2 или 5 (в симметричном случае $n-1$ или $n-4$), а в конце колышек может остаться на клетке $n-1$ или $n-4$ (в симметричном случае 2 или 5).

Решение вытекает из анализа ситуации, которая возникает после любого первого хода

$$\underbrace{1\ldots1}_{l}00\underbrace{1\ldots1}_{r},$$

где единицы и нули обозначают клетки с колышком и без соответственно. Без ограничения общности можно считать, что $l \leqslant r$.

Если $l=0$ и $r=2$, то после единственного возможного хода последним колышком он останется на доске один; если $l=0$ и $r>2$, то после $\left\lfloor \frac{r}{2} \right\rfloor$ обязательных прыжков влево останутся $\left\lceil \frac{r}{2} \right\rceil \geqslant 2$ колышков, разделённых пустыми клетками.

Точно так же, если $l=1$ и $r \geqslant 1$, после $\left\lfloor \frac{r}{2} \right\rfloor$ обязательных прыжков влево останутся $\left\lceil \frac{r}{2} \right\rceil + 1 \geqslant 2$ колышков, разделённых пустыми клетками.

Если $l=2$, а $r \geqslant 2$ и чётно, то все колышки, кроме одного, можно убрать после некоторого числа разрешённых прыжков — это легко показать с помощью индукции по r. Действительно, если $r=2$, то цель достигается после прыжков первого, последнего и затем одного из двух оставшихся колышков:

$$110011 \implies 001011 \implies 001100 \implies \text{либо } 010000, \text{ либо } 000010.$$

С помощью таких же двух первых прыжков задача для чётного числа колышков $r>2$ сводится к задаче для $r-2$

колышков, что доказывает шаг индукции:

$$110\underbrace{011\ldots1}_{r} \implies 001\underbrace{011\ldots1}_{r} \implies 0011001\underbrace{1\ldots1}_{r-2}.$$

Более того, поскольку нельзя решить задачу после ситуации $0\underbrace{11\ldots1}_{r}$, где $r > 2$, существует лишь один способ решить задачу после $1100\underbrace{11\ldots1}_{r}$ — вышеописанный.

Наконец, если $l > 2$ и $r \geqslant l$, оставить на доске только один колышек невозможно. Действительно, ни из $l > 2$ колышков слева от 00, ни из $r > 2$ колышков справа нельзя получить один колышек, не прибегая к «помощи» колышка с другой стороны. Этот колышек можно «привлечь», если он перепрыгнет с другой стороны через своего соседа, но один колышек нельзя получить таким образом:

$$\underbrace{1\ldots11}_{l}00\underbrace{11\ldots1}_{r} \implies \ldots \implies \underbrace{1\ldots1}_{l-2}001100\underbrace{1\ldots1}_{r-2}.$$

Комментарии. Главный инструмент в решении этой головоломки — стратегия «уменьшай и властвуй». Используя такой же подход, можно также показать, что единственной доской с нечётным числом клеток, для которой задача имеет решение, является доска размера 3, с пустой клеткой либо 1, либо 3. В статье [Moo00] приведено формальное описание всех позиций, которые могут появиться в этой игре.

Эта головоломка — одномерная версия старой, но до сих пор популярной игры, в которую играют с теми же правилами, но на двумерной доске. Глубокий разбор истории игры и выигрышных стратегий см., например, в монографии [Bea92] и в гл. 23 книги [Ber04], а также на веб-сайтах, посвящённых этой игре.

118. Шесть коней

Решение. Шестнадцать — минимальное число ходов, которые нужно сделать для решения задачи.

Как объясняется в первой части учебного раздела, начальное расположение коней в этой головоломке легко представить в виде графа, изображённого на рис. 4.81.

Граф можно развернуть так, как показано на рис. 4.82: переместим вершину 8 наверх и вершину 5 вниз, чтобы получить первый граф на этом рисунке; поменяем местами

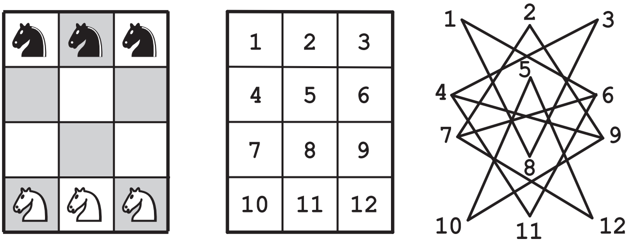


Рис. 4.81. Граф, показывающий возможные ходы в задаче «*Шесть коней*»

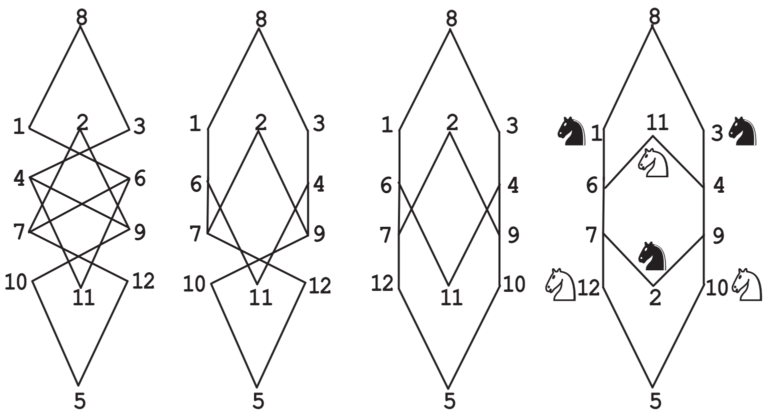


Рис. 4.82. Разворачивание графа на рис. 4.81

вершины 4 и 6, чтобы получить второй; поменяем местами вершины 10 и 12, чтобы получить третий; переместим вершину 11 наверх и вершину 2 вниз, чтобы получить развёрнутое представление нашего графа.

Чтобы достичь цели задачи, одного из двух чёрных коней с начальных позиций 1 и 3 нужно передвинуть либо на 12, либо на 10. Без ограничения общности мы будем считать, что конь с клетки 1 перемещается на клетку 12, которая к его начальной позиции ближе, чем 10. Если воспользоваться кратчайшим путём, это займёт три хода. Для того чтобы переместить двух других чёрных коней с начальных позиций 2 и 3 на 10 и 11, потребуется всего как минимум четыре хода.

Из соображений симметрии те же минимум семь ходов понадобится сделать, чтобы переместить на требуемые позиции белых коней. Таким образом, головоломку нельзя решить меньше чем за 14 ходов. Однако это нельзя сделать ни за 14, ни за 15 ходов, что легко доказать от противного. Предположим, что есть последовательность ходов, приводящая к решению задачи меньше чем за 16 ходов. В такой последовательности чёрный конь с клетки 1 должен был бы достичь клетки 12 за три хода, поскольку в четыре хода это сделать невозможно, а если бы он делал пять или больше ходов, то общее число ходов равнялось бы как минимум $(5+4)+7=16$.

Но прежде чем чёрный конь с 1 сможет пойти в 6 или 7, белый конь с 12 должен пойти в 2, а перед этим чёрный конь с 2 должен переместиться в 9, а до этого белый конь с 10 должен пойти в 4, а до этого чёрный конь с 3 должен пойти в 11, а до этого белый конь с 11 должен пойти в 6, преградив путь чёрному коню на 1 — тем самым, необходимое число ходов становится бо́льшим 15.

В то же время головоломку можно решить за 16 ходов, например с помощью такой последовательности (B — чёрный конь, W — белый конь):

$$B(1-6-7),\ W(11-6-1),\ B(3-4-11),\ W(10-9-4-3),$$
$$B(2-9-10),\ B(7-6),\ W(12-7-2),\ B(6-7-12).$$

Комментарии. Эта головоломка — известное обобщение головоломки Гуарини, которая обсуждалась в первой части учебного раздела. Основная идея в обеих головоломках — изменение представления: сначала мы представляем доску как граф, а затем разворачиваем граф, чтобы получить более ясную картину нашей задачи.

Journal of Recreational Mathematics опубликовал две статьи, содержащие неправильные ответы к этой задаче, в 1974 и 1975 гг. [Sch80, p. 120–124]. Дьюдени [Dud02, задача 94] рассматривал эту задачу с дополнительным ограничением: ни на каком ходу два коня противоположного цвета не должны оказаться в позициях, из которых они могут напасть друг на друга. Другие варианты см. в книгах [Pet97, p. 57–58]) и [Gra05, p. 204–206].

119. Замощение цветными тримино

Решение. Эту головоломку можно решить с помощью следующего рекурсивного алгоритма. Если $n=2$, разделим доску на четыре доски 2×2 и положим одно серое тримино так, чтобы покрыть те три центральных квадрата, которые не находятся на одной доске 2×2 с выброшенной клеткой. Затем поместим одно чёрное тримино в левую верхнюю доску 2×2, одно белое

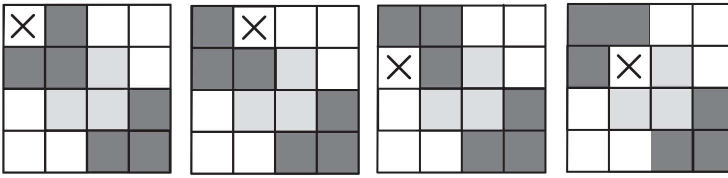


Рис. 4.83. Покрытие цветными тримино доски 4×4 для четырёх возможных позиций выброшенной клетки (обозначена как X)

тримино в правую верхнюю доску 2×2, одно чёрное тримино в правую нижнюю доску 2×2 и одно белое тримино в левую нижнюю доску 2×2 (см. рис. 4.83).

Заметим, что, какая бы клетка ни отсутствовала, имеют место следующие свойства:

- Верхний край доски, если идти слева направо, покрыт повторяющейся последовательностью из двух чёрных квадратов и затем двух белых квадратов, причём один из квадратов последовательности, возможно, заменяет выброшенный квадрат.
- Правый край доски, если идти сверху вниз, покрыт повторяющейся последовательностью из двух белых квадратов и затем двух чёрных квадратов, причём один из квадратов последовательности, возможно, заменяет выброшенный квадрат.
- Нижний край доски, если идти справа налево, покрыт повторяющейся последовательностью из двух чёрных квадратов и затем двух белых квадратов, причём один из квадратов последовательности, возможно, заменяет выброшенный квадрат.
- Левый край доски, если идти снизу вверх, покрыт повторяющейся последовательностью из двух белых квадратов и затем двух чёрных квадратов, причём один из квадратов последовательности, возможно, заменяет выброшенный квадрат.

Если $n > 2$, разделим доску на четыре доски $2^{n-1} \times 2^{n-1}$ и поместим одно серое тримино так, чтобы оно покрыло те три центральных квадрата, которые не находятся на одной доске $2^{n-1} \times 2^{n-1}$ с выброшенной клеткой. Затем покроем каждую из трёх досок $2^{n-1} \times 2^{n-1}$, используя описанный выше алгоритм рекурсивно (см. пример на рис. 4.84).

Доказательство корректности алгоритма основано на том факте, что вышеперечисленные свойства верны на каждой

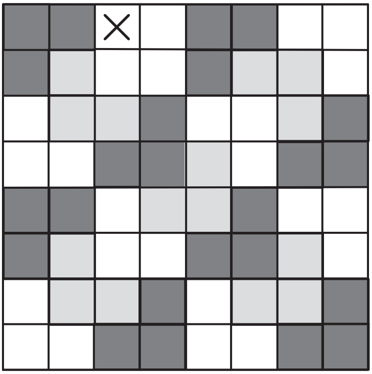


Рис. 4.84. Покрытие цветными тримино доски $2^3 \times 2^3$ с выброшенным квадратом (обозначен через X)

итерации алгоритма, а это легко проверить с помощью математической индукции.

Комментарии. Как и в случае алгоритма для покрытия неокрашенными тримино, который обсуждался в первой части учебного раздела, этот — превосходная иллюстрация пользы стратегии «разделяй и властвуй».

И головоломка, и её решения взяты из статьи [Chu87].

120. Машина, распределяющая пенни

Решение. а) Будем считать, что коробки пронумерованы слева направо, начиная с нуля, который обозначает самую левую коробку. Пусть $b_0 b_1 \ldots b_k$ — строка, обозначающая результат распределения машиной n пенни, где b_i равняется 1 или 0 в зависимости от того, есть в i-й, $0 \leqslant i \leqslant k$, коробке монета или нет; в частности, $b_k = 1$, где k — номер последней коробки с пенни. Эта монета оказалась там, заменив две монеты из коробки $k-1$, которые в свою очередь заменили четыре монеты из коробки $k-2$, и т. д.

Применяя подобные рассуждения к любой коробке i, в которой в конце есть монета, получим формулу

$$n = \sum_{i=0}^{k} b_i 2^i.$$

Другими словами, битовая строка, обозначающая финальное распределение, есть также записанное в обратном порядке двоичное представление начального числа пенни n. Поскольку двоичное представление любого натурального числа единственно, машина всегда заканчивает работу с одним и тем же распределением монет для данного n, вне зависимости от порядка, в котором она обрабатывает пары монет.

б) Минимальное число коробок, которое требуется, чтобы распределить n пенни, равняется количеству цифр в двоичном представлении числа n, т. е. $\lfloor \log_2 n \rfloor + 1 = \lceil \log_2 (n+1) \rceil$.

в) Пусть $b_k b_{k-1} \ldots b_0$ есть двоичное представление числа n. Согласно ответу из части а), когда машина останавливается, число пении в коробке i равняется $b_i, 0 \leqslant i \leqslant k$. Имеем следующее рекуррентное соотношение для числа итераций $C(i)$, которые должна совершить машина, чтобы поместить один пенни в коробку i:

$$C(i) = 2C(i-1) + 1 \quad \text{при } 0 < i \leqslant k, \quad C(0) = 0.$$

Решая соотношение методом обратных подстановок, получим следующее:

$$\begin{aligned} C(i) &= 2C(i-1) + 1 = \\ &= 2(2C(i-2) + 1) + 1 = 2^2 C(i-2) + 2 + 1 = \\ &= 2^2(2C(i-3) + 1) + 2 + 1 = 2^3 C(i-3) + 2^2 + 2 + 1 = \\ &= \ldots = \\ &= 2^i C(i-i) + 2^{i-1} + 2^{i-2} + \cdots + 1 = 2^i \cdot 0 + (2^i - 1) = 2^i - 1. \end{aligned}$$

Следовательно, общее число итераций, которые совершит машина, прежде чем остановится, можно найти как

$$\sum_{i=0}^{k} b_i C(i) = \sum_{i=0}^{k} b_i (2^i - 1) = \sum_{i=0}^{k} b_i 2^i - \sum_{i=0}^{k} b_i = n - \sum_{i=0}^{k} b_i,$$

что есть разность между начальным числом пенни и числом единиц в его двоичном представлении.

Комментарии. В этой головоломке используется двоичное представление числа n, которое в данном случае является инвариантом, и решение «от конца к началу». Эта задача — несколько изменённая версия задачи, которую придумал Джеймс Пропп (см. [MathCircle]).

121. Проверка супер-яйца

Решение. Ответом является 14.

Пусть $H(k)$ — максимальное число этажей, для которых задача может быть решена за k бросков. Первый раз бросить яйцо нужно с этажа k, потому что если оно разобьётся, то все нижние $k-1$ этажей необходимо будет проверить последовательно, начиная с первого. Если же после первого броска яйцо не разбилось, то второй следует сделать с этажа $k + (k-1)$ и быть готовым к тому, что, если яйцо разобьётся, нужно будет последовательно проверить все этажи с $k+1$ до $2k-2$.

Повторяя эти рассуждения для оставшихся $k-2$ бросков, получаем следующую формулу для $H(k)$:

$$H(k) = k + (k-1) + \cdots + 1 = \frac{k(k+1)}{2}.$$

(Эту же формулу можно получить и по-другому, решив рекуррентное соотношение $H(k) = k + H(k-1)$ для $k > 1$, $H(1) = 1$.)

Чтобы ответить на вопрос задачи, осталось найти наименьшее k такое, что $\frac{k(k+1)}{2} \geqslant 100$. Таким является $k = 14$. Первое яйцо нужно бросать с этажей 14, 27, 39, 50, 60, 69, 77, 84, 90, 95, 99 и 100, пока оно не разобьётся, и, если это произойдёт, второе яйцо нужно будет бросать, раз за разом поднимаясь на один этаж, начиная со следующего за тем, который последним был проверен успешно. Это решение не единственно: первый бросок также можно совершить с этажей 13, 12, 11 и 10, если, конечно, соответственно изменить остальные броски.

Комментарии. В то время как алгоритм решения задачи, очевидно, основан на применении жадного подхода, несколько необычным является анализ худшего случая наоборот: алгоритм ищет максимальный размер задачи для данного числа выполнений его основной операции (бросание яйца).

После того как эта головоломка появилась в книге [Kon96, Problem 166], она стала весьма популярной: см., например, книгу [Win07, p. 10] и статью [Sni03].

122. Мир в парламенте

Решение. Ответ — «правда»: ниже мы приводим алгоритм разделения парламента на две палаты так, чтобы ни у какого парламентария не было больше одного врага в его/её палате.

Начнём с того, что разделим парламентариев на две палаты произвольным образом (например, примерно поровну). Пусть p — общее в двух палатах число пар врагов, которые на данный момент находятся в одной палате. Если есть парламентарий, у которого минимум двое врагов в его палате, переместим его в другую палату. Поскольку в новой палате у этого парламентария будет не больше одного врага, а число пар врагов в его старой палате уменьшится на два, общее число пар врагов уменьшится по меньшей мере на один. Следовательно, процесс остановится после не более чем p итераций, и в конце ни у какого парламентария не будет больше одного врага в его/её палате.

Комментарии. Можно считать алгоритм решения этой головоломки основанным на стратегии итерационного улучшения, упомянутой в первой части учебного раздела. Подробное обсуждение этой стратегии, а также нескольких важных алгоритмов, которые её используют, можно найти в книге А. Левитина [Lev06, гл. 10].

Самое раннее упоминание этой головоломки, которое мы смогли найти, относится к августу 1979 года — задача M580 (с. 38) в соответствующем выпуске «Кванта», научно-популярного журнала по физике и математике для школьников и учителей. Затем она была включена и в некоторые книги на английском языке, см. [Sav03, p. 1, Problem 4].

123. Задача о флаге Нидерландов

Решение. Нижеизложенный алгоритм основан на том, что ряд шашек рассматривается как состоящий из четырёх частей, которые могут быть и пустыми: слева красные шашки, затем белые, затем шашки пока неопределённого цвета и в конце синие шашки:

красные	белые	неопределённые	синие

Изначально мы считаем красные, белые и синие части пустыми — все шашки лежат в неопределённой части. На каждом шаге алгоритм уменьшает неопределённость на одну шашку слева или справа: если первая (т. е. самая левая) шашка в неопределённой части красная, меняем её местами с первой шашкой после красной части и переходим к следующей шашке; если она белая, переходим к следующей шашке; если она синяя, поменяем её местами с последней шашкой перед синей частью. (Алгоритм иллюстрирует рис. 4.85.) Операция повторяется до тех пор, пока неопределённая часть не опустеет.

Комментарии. И задача, и алгоритм были предложены В. Х. Дж. Фейеном; известной в информатике она стала благодаря выдающемуся учёному Эдсгеру Дейкстре: он включил её в свою книгу «Дисциплина программирования» [Dij76, ч. 14]. Название задачи легко объяснить: и Фейен, и Дейкстра были нидерландцами, а красный, белый и синий — действительно цвета флага Нидерландов.

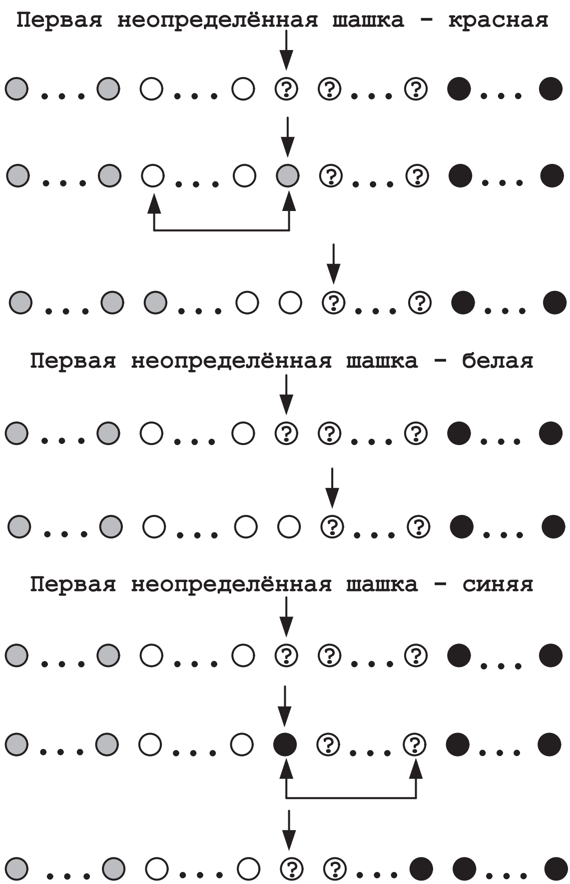


Рис. 4.85. Три случая в алгоритме решения головоломки «*Задача о флаге Нидерландов*»

124. Разделение цепочки

Решение. Ответом является наименьшее целое k такое, что $(k+1)\,2^{k+1}-1 \geqslant n$.

Как и предлагается в подсказке к этой задаче, легче ответить на обратный вопрос: найти сначала $n_{\max}(k)$ — максимальную длину цепи, для которой задачу можно решить удалением k одиночных скрепок, где k есть данное положительное целое число. Если убрать k скрепок из разных мест цепи, она разобьётся на $k+1$ частей. Самая короткая из них, S_1, должна быть длиной в $k+1$ скрепок: это позволит нам, пользуясь ей и убранными k скрепками, составить цепь любой длины от 1 до $k + (k+1) = 2k+1$. Вторая из самых коротких частей, S_2, должна быть на одну скрепку длиннее, т. е. состоять из $(2k+1)+1 = 2(k+1)$ скрепок. С ней мы сможем составить цепь любой длины от 1 до $(2k+1)+2(k+1) = 4k+3$. Третья из самых коротких частей, S_3, должна быть длиной $(4k+3)+1 = 2^2(k+1)$, и т. д., пока мы не дойдём до самой длинной части S_{k+1} длиной $2^k(k+1)$. (Это утверждение можно формально доказать с помощью математической индукции.)

Если у нас есть k убранных скрепок и эти $k+1$ частей, мы можем составить цепь любой целой длины от 1 до $n_{\max}(k)$ включительно, где

$$\begin{aligned} n_{\max}(k) &= k + (k+1) + 2(k+1) + 2^2(k+1) + \cdots + 2^k(k+1) = \\ &= k + (k+1)(1 + 2 + 2^2 + \cdots + 2^k) = k + (k+1)(2^{k+1} - 1) = \\ &= (k+1)2^{k+1} - 1. \end{aligned}$$

Например, при $k = 2$ мы можем собрать цепь любой длины от 1 до 23, как показано на следующей схеме:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
			X							X												

Вернёмся теперь к задаче в исходной формулировке, и пусть $n > 1$ — длина данной цепи. Очевидно, что существует единственное положительное целое k такое, что

$$n_{\max}(k-1) < n \leqslant n_{\max}(k),$$

где $n_{\max}(k) = (k+1)\,2^{k+1} - 1$. Из вышеизложенного мы знаем: чтобы решить задачу, нужно удалить из цепи по меньшей мере k одиночных скрепок. Теперь покажем, что для нашей цели, на самом деле, достаточно убрать k скрепок описанным выше способом с небольшими изменениями, которые мы поясним ниже. Если $n = n_{\max}(k)$, то, убрав k одиночных скрепок так, как описано выше, мы решим задачу. Если $|S_1| + \ldots + |S_k| + k \leqslant n < n_{\max}(k)$, где $|S_1|, \ldots, |S_k|$ обозначают длины первых k частей цепи, которые мы получили так же, как выше, то последняя часть, $\tilde{S}_{k+1}$, будет короче, чем последняя часть S_{k+1} в решении выше. Но легко видеть, что такое разбиение данной цепи по-прежнему даёт решение задачи. Например, если $n = 20$ и, следовательно, $k = 2$, то для только что описанного решения нужно убрать скрепки 4 и 11:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			X							X									

Цепь любой длины от 1 до $|S_1| + |S_2| + 2 = 3 + 6 + 2 = 11$ можно получить, комбинируя части S_1, S_2 и две одиночные скрепки; значит, вместе с $\tilde{S}_3$ можно также получить цепь любой длины от $\left|\tilde{S}_3\right| = 9$ до $\left|\tilde{S}_3\right| + 11 = 20$.

Если $n_{\max}(k-1) < n < |S_1| + \ldots + |S_k| + k$, то k-ю одиночную скрепку можно убрать из правого конца данной цепи; в то же время первые $k-1$ одиночных скрепок убираются из тех же позиций, что и выше. Например, если $n = 10$ и, следовательно, $k = 2$, то скрепки нужно убрать из позиций 4 и 10:

1	2	3	4	5	6	7	8	9	10
			X						X

Цепь любой длины от 1 до $|S_1| + 2 = 3 + 2 = 5$ можно получить, комбинируя часть S_1 и две одиночные скрепки; значит, вместе с частью $\tilde{S}_2$ длины 5 можно также получить цепь любой длины от 5 до 10.

Комментарии. Вышеприведённое решение (см., например, [Sch80, p. 128–130]) хорошо иллюстрирует жадную стратегию в применении к разработке алгоритмов.

В самой известной версии этой головоломки речь идёт о золотой семизвенной цепи, которую путешественник должен разрезать, чтобы заплатить хозяину таверны за каждый день недели, которую проводит в таверне. Несколько связанных с задачей ссылок можно найти в аннотированной библиографии Д. Сингмастера [Sin10, Section 5.S.1].

125. Отсортировать 5 за 7

Решение. Полезно начать с очевидной, но не очень плодотворной идеи: отсортировать по весу сначала четыре предмета, а затем найти среди них место для пятого. Отсортировать четыре числа путём сравнения можно не менее чем за пять шагов. (Нетрудно доказать это напрямую; к тому же это следует из общей теоремы о том, что сортировка n произвольных действительных чисел требует по меньшей мере $\lceil \log_2 n! \rceil$ шагов (сравнений) в наихудшем случае.) Далее легко заметить, что для нахождения места пятого предмета нужно по меньшей мере ещё три взвешивания. (Это становится ясным, если рассмотреть отсортированные числа на вещественной прямой.)

Вот алгоритм для решения задачи. Пронумеруем предметы произвольно от 1 до 5. Пусть $w_1, \ldots, w_5$ — это неизвестные веса предметов. Сначала сравним вес предметов 1 и 2, а также 3 и 4. Без ограничения общности мы можем предположить, что $w_1 < w_2$ и $w_3 < w_4$ Взвесим два более тяжёлых предмета, т. е. сравним w_2 и w_4 После этих трёх взвешиваний могут быть два варианта:

$$\text{Вариант 1: } w_1 < w_2 < w_4 \quad \text{и } w_3 < w_4,$$
$$\text{Вариант 2: } w_3 < w_4 < w_2 \quad \text{и } w_1 < w_2.$$

Очевидно, что вариант 2 аналогичен варианту 1, поскольку они отличаются только ролью, которую играют первая и вторая пары весов при первых двух взвешиваниях. Поэтому без ограничения общности достаточно рассмотреть вариант 1. Как мы уже сказали, полезно представить числа, которые надо отсортировать, на вещественной прямой. Таким образом, после первых трёх взвешиваний получим такую диаграмму:

$$-w_1 - w_2 - w_4 - \ \text{ и } w_3 < w_4 \text{ (точка } w_3 \text{ левее } w_4).$$

При четвёртом взвешивании сравним w_5 и w_2. Если $w_5 < w_2$, пятым взвешиванием сравним w_5 и w_1. Получаем одну из двух ситуаций:

$$\text{Если } w_5 < w_1\text{: } -w_5 - w_1 - w_2 - w_4 - \ \text{ и } w_3 < w_4 \ (w_3 \text{ левее } w_4).$$
$$\text{Если } w_5 > w_1\text{: } -w_1 - w_5 - w_2 - w_4 - \ \text{ и } w_3 < w_4 \ (w_3 \text{ левее } w_4).$$

Поскольку эти две ситуации отличаются только относительным расположением w_1 и w_5, мы далее проведём сравнения только для первой ситуации. Шестым взвешиванием сравним w_3 и w_1. Если $w_3 < w_1$, то седьмым взвешиванием сравним w_3 и w_5. Если $w_3 < w_5$, получаем $w_3 < w_5 < w_1 < w_2 < w_4$. Если же $w_3 > w_5$, получаем $w_5 < w_3 < w_1 < w_2 < w_4$. Аналогично, если при шестом взвешивании $w_3 > w_1$, то седьмым взвешиванием сравниваем w_3 и w_2. Если $w_3 < w_2$, получаем $w_5 < w_1 < w_3 < w_2 < w_4$; если $w_3 > w_2$, получаем $w_5 < w_1 < w_2 < w_3 < w_4$.

Теперь рассмотрим ситуацию, когда $w_5 > w_2$ при четвёртом взвешивании. Тогда пятым взвешиванием сравним w_5 и w_4 и получим один из двух вариантов:

$$\text{Если } w_5 < w_4\text{: } -w_1 - w_2 - w_5 - w_4 - \ \text{ и } w_3 < w_4 \ (w_3 \text{ слева от } w_4).$$
$$\text{Если } w_5 > w_4\text{: } -w_1 - w_2 - w_4 - w_5 - \ \text{ и } w_3 < w_4 \ (w_3 \text{ слева от } w_4).$$

В первом варианте шестым взвешиванием сравним w_3 и w_2. Если $w_3 < w_2$, седьмым взвешиванием сравниваем w_3 и w_1. Если $w_3 < w_1$, получаем $w_3 < w_1 < w_2 < w_5 < w_4$; если $w_3 > w_1$, получаем $w_1 < w_3 < w_2 < w_5 < w_4$. При втором варианте шестым взвешиванием тоже сравним w_3 и w_2. Если $w_3 < w_2$, то седьмым взвешиванием сравним w_3 и w_1. Если $w_3 < w_1$, получаем $w_3 < w_1 < w_2 < w_4 < w_5$; если же $w_3 > w_1$, получаем $w_1 < w_3 < w_2 < w_4 < w_5$. Наконец, если $w_3 > w_2$ при шестом взвешивании, то седьмое взвешивание не нужно, так как мы знаем, что $w_3 < w_4$. Поэтому мы получаем $w_1 < w_2 < w_3 < w_4 < w_5$.

Комментарии. Головоломка представляет собой хорошо известную задачу сортировки с наименьшим числом сравнений. Дональд Кнут обсуждал её в своей книге «Искусство программирования» [Knu98], в разд. 5.3.1, том 3. В частности, в книге есть элегантная диаграмма, показывающая этот алгоритм, разработанный Г. Б. Демутом (p. 183–184).

126. Деление пирога по-честному

Решение. Честное решение для двух человек — один режет пирог на два куска, а второй выбирает кусок себе. Если людей больше двух, можно применить такую процедуру. Сначала присвоим всем людям номера от 1 до n. Человек 1 отрезает от пирога кусочек X. Он старается отрезать кусок размером $\frac{1}{n}$. Если он отрежет кусок поменьше, то он его и получит. Если побольше, то остальные, скорее всего, уменьшат этот кусок. Если человек 2 считает, что кусок X больше, чем $\frac{1}{n}$, то он может отрезать кусочек от X и положить его обратно к пирогу. Если человек 2 так не считает, то он ничего не делает. Люди $3, 4, \ldots, n$ по очереди делают то же самое — либо уменьшают кусок X, либо не делают ничего. В итоге последний человек берёт X себе, а оставшиеся $n - 1$ людей продолжают ту же процедуру, чтобы разделить оставшуюся часть пирога, в том числе все лишние кусочки, отрезанные от X (если таковые имеются). Когда остаются всего два человека, они делят оставшуюся часть (с кусочками) — один отрезает, а другой выбирает кусок.

Комментарии. Алгоритм, очевидно, основан на стратегии «уменьшай на единицу». Как отметил Иэн Стюарт [Ste06, p. 4–5], решение задачи для двух людей известно в течение

2800 лет, а решение усложнённого варианта для трёх людей было найдено польским математиком Гуго Штейнгаузом в 1944 г. С тех пор для решения первоначальной версии задачи и её вариаций было предложено несколько различных алгоритмов. Те, кому это интересно, могут найти их в монографии [Rob98].

127. Задача о ходе коня

Решение. Поскольку доска симметрична, мы можем без ограничения общности начать маршрут коня в верхнем левом углу доски и закончить его на клетке с номером 64 на рис. 4.86. Будем ходить всегда на клетку, на которой ещё не были и которая расположена как можно ближе к ближайшему краю доски. Точнее, мы попытаемся сначала пройти по клеткам двух внешних рядов клеток, перепрыгивая на 16 центральных клеток только в крайнем случае.

Кроме того, будем ходить на угловую клетку, как только это станет возможным. Маршрут, составленный по этим правилам, показан на рис. 4.86. Клетки на рисунке пронумерованы в той последовательности, в которой по ним ходит конь. Обратите внимание, что на двенадцатом ходу мы вынуждены пойти на одну из центральных клеток, чтобы не попасть на клетку 64, которая должна быть окончанием маршрута.

Комментарии. Задача о ходе коня — одна из двух наиболее изученных головоломок про шахматную доску (вторая — это задача об n ферзях, о которой говорится в первой части учебного раздела и в головоломке № 140). В аннотированной библиографии Д. Сингмастера этой головоломке посвящён целый раздел в восемь страниц [Sin10, Section 5.F.1], а её история

1	38	17	34	3	48	19	32
16	35	2	49	18	33	4	47
39	64	37	54	59	50	31	20
36	15	56	51	62	53	46	5
11	40	63	60	55	58	21	30
14	25	12	57	52	61	6	45
41	10	27	24	43	8	29	22
26	13	42	9	28	23	44	7

Рис. 4.86. Замкнутый маршрут коня на обычной шахматной доске

прослеживается с IX века. Всё это время задача привлекала внимание известных математиков, в том числе великих Леонарда Эйлера и Карла Гаусса. На доске размером 8×8 можно проложить более чем 10^{13} замкнутых маршрутов коня, поэтому удивительно, что нахождение хотя бы одного из них без помощи компьютера — нетривиальная задача.

Описанный нами маршрут основан на идее, предложенной в начале XVIII века Пьером де Монмором и Абрахамом де Муавром для построения открытого маршрута, т. е. маршрута, который не должен заканчиваться на исходной клетке. Идея основана на жадном подходе, при котором в первую очередь «захватываются» клетки, на которые можно попасть с меньшего числа клеток. Этот подход использовался более формально в методе, предложенном на век позже Варнсдорфом: из возможных клеток выбирай ту, с которой меньше всего следующих возможных ходов.

Метод Варнсдорфа является более плодотворным из этих двух жадных методов, но для него требуется больше расчётов. Важно отметить, что оба эти метода всего лишь эвристические: логические правила могут и не привести к решению (в нашем случае из-за обрыва пути). В общем случае идея эвристических алгоритмов плодотворна для сложных расчётных задач. Другие алгоритмы для прокладывания маршрута коня (некоторые из них основаны на варианте стратегии «разделяй и властвуй») описаны, например, в [Bal87, p. 175–186], [Kra53, p. 257–266], [Gik76, с. 51–67], а также на многочисленных веб-сайтах, посвящённых этой задаче и её усложнённым версиям.

Стоит также отметить, что задача о ходе коня — это частный случай нахождения гамильтонова цикла, а эвристический подход Варнсдорфа для этого как раз часто и используется.

128. Тумблеры системы охраны

Решение. Минимальное необходимое число переключений для решения головоломки $\frac{2}{3}2^n - \frac{1}{6}(-1)^n - \frac{1}{2}$. Пронумеруем тумблеры слева направо от 1 до n и примем состояние «включено» за 1, а состояние «выключено» за 0. Чтобы легче было решить общий случай задачи, начнём с решения четырёх простых частных случаев (см. рис. 4.87).

Теперь рассмотрим общий случай; в начале состояние тумблеров задаётся двоичной строкой из n единиц: 111...1. Прежде чем мы сможем выключить первый (самый левый)

$n = 1$

1
0

$n = 2$

1	1
0	1
0	0

$n = 3$

1	1	1
1	1	0
0	1	0
0	1	1
0	0	1
0	0	0

$n = 4$

1	1	1	1
1	1	0	1
1	1	0	0
0	1	0	0
0	1	0	1
0	1	1	1
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1
0	0	0	0

Рис. 4.87. Оптимальное решение для первых четырёх частных случаев задачи «*Тумблеры системы охраны*»

тумблер, нужно добиться состояния 110…0. Следовательно, если мы хотим разработать оптимальный алгоритм, для начала нужно за минимальное число ходов выключить последние $n-2$ тумблеров. Другими словами, для начала нужно решить ту же задачу для последних $n-2$ тумблеров. Это можно сделать рекурсивно, причём случаи $n = 1$ и $n = 2$ решаются напрямую, как показано на рис. 4.87. После этого мы сможем переключить первый тумблер и получить 010…0. Теперь, прежде чем мы сможем переключить второй тумблер, нам придётся пройти состояние, в котором включены все последующие тумблеры, — это легко доказать с помощью математической индукции. Включить все тумблеры от третьего до последнего можно с помощью последовательности ходов, обратной к той, оптимальной, с помощью которой мы ранее перевели все последние $n-2$ тумблеров из состояния «включено» в состояние «выключено». Так мы придём к 011…1. Не считая первого 0, имеем теперь ту же задачу, но для $n-1$, и её можно решить рекурсивно.

Пусть $M(n)$ — число ходов (переключений), которое делает вышеописанный алгоритм. Имеем следующее рекуррентное соотношение для $M(n)$:

$$M(n) = M(n-2) + 1 + M(n-2) + M(n-1),$$

или

$$M(n) = M(n-1) + 2M(n-2) + 1 \quad \text{при} \quad n \geqslant 3,$$
$$M(1) = 1, \quad M(2) = 2.$$

Решив это соотношение с помощью обычной техники решения неоднородных линейных рекуррентных соотношений второго порядка с постоянными коэффициентами (см., например, [Lev06, p. 476–478] или [Ros07, Section 7.2]), получим следующее решение:

$$M(n) = \frac{2}{3}\,2^n - \frac{1}{6}\,(-1)^n - \frac{1}{2} \qquad \text{при} \quad n \geqslant 1.$$

При чётных n эту формулу можно упростить до $M(n) = \dfrac{2^{n+1}-2}{3}$; при нечётных n — до $M(n) = \dfrac{2^{n+1}-1}{3}$.

Комментарии. Алгоритм решения головоломки основан на стратегии «уменьшай и властвуй». Несмотря на то, что решить линейное рекуррентное соотношение второго порядка для числа ходов с помощью стандартной техники и естественно, и легко, этого можно избежать, следуя советам в книгах [Bal87, p. 318–320] и [Ave00, p. 414]. С нашей точки зрения, эти методы более громоздкие, чем вышеприведённое решение. Совершенно другой подход предложил французский математик Луи А. Гро в 1872 г. Его метод не отличался от представления тумблеров в виде двоичных строк и предвосхитил возникновение современных кодов Грэя; подробнее см. [Bal87, p. 320–322] и [Pet09, p. 182–184].

Эта головоломка была предложена в статье [Gre73]. Она имитирует очень старую и известную механическую головоломку «Китайские кольца». Обширный список литературы о Китайских кольцах приведён Д. Сингмастером [Sin10, Section 7.M.1]; несколько более современных ссылок можно найти в [Pet09, p. 182].

129. Головоломка Реве

Решение. Поскольку задача, очевидно, является обобщением головоломки «Ханойская башня», естественно использовать тот же рекурсивный подход. А именно, если $n > 2$, рекурсивно переместим k самых маленьких колец на промежуточный стержень, используя все четыре стержня. Затем переместим оставшиеся $n-k$ колец на тот стержень, на который требуется, с помощью классического рекурсивного алгоритма для Ханойской башни в случае трёх стержней (см. втор часть учебного раздела). Наконец, рекурсивно переместим k самых маленьких колец на требуемый стержень, используя все четыре стержня.

Тривиальные случаи задачи $n = 1$ или 2 решаются, соответственно, в один и три хода, так же, как в решении задачи о Ханойской башне для трёх стержней. Значение параметра k нужно выбрать таким, чтобы минимизировать общее число перемещений колец. Это приводит нас к следующему рекуррентному соотношению для числа шагов алгоритма $R(n)$, которые нужно сделать, чтобы переместить n колец:

$$R(n) = \min_{1 \leqslant k < n} [2R(k) + 2^{n-k} - 1] \quad \text{при } n > 2,\ R(1) = 1,\ R(2) = 3.$$

Начав с $R(1) = 1$ и $R(2) = 3$ и используя это соотношение, можно найти значения $R(3), R(4), \ldots, R(8)$. Эти значения выделены жирным шрифтом в таблице ниже.

n	k	$2R(k) + 2^{n-k} - 1$	n	k	$2R(k) + 2^{n-k} - 1$
3	1	$2 \cdot 1 + 2^2 - 1 = \mathbf{5}$	5	1	$2 \cdot 1 + 2^4 - 1 = 17$
	2	$2 \cdot 3 + 2^1 - 1 = 7$		2	$2 \cdot 3 + 2^3 - 1 = \mathbf{13}$
				3	$2 \cdot 5 + 2^2 - 1 = \mathbf{13}$
4	1	$2 \cdot 1 + 2^3 - 1 = \mathbf{9}$		4	$2 \cdot 9 + 2^1 - 1 = 19$
	2	$2 \cdot 3 + 2^2 - 1 = \mathbf{9}$			
	3	$2 \cdot 5 + 2^1 - 1 = 11$			

n	k	$2R(k) + 2^{n-k} - 1$	n	k	$2R(k) + 2^{n-k} - 1$
6	1	$2 \cdot 1 + 2^5 - 1 = 33$	7	1	$2 \cdot 1 + 2^6 - 1 = 65$
	2	$2 \cdot 3 + 2^4 - 1 = 21$		2	$2 \cdot 3 + 2^5 - 1 = 37$
	3	$2 \cdot 5 + 2^3 - 1 = \mathbf{17}$		3	$2 \cdot 5 + 2^4 - 1 = \mathbf{25}$
	4	$2 \cdot 9 + 2^2 - 1 = 21$		4	$2 \cdot 9 + 2^3 - 1 = \mathbf{25}$
	5	$2 \cdot 13 + 2^1 - 1 = 27$		5	$2 \cdot 13 + 2^2 - 1 = 29$
				6	$2 \cdot 17 + 2^1 - 1 = 35$

n	k	$2R(k) + 2^{n-k} - 1$
8	1	$2 \cdot 1 + 2^7 - 1 = 129$
	2	$2 \cdot 3 + 2^6 - 1 = 69$
	3	$2 \cdot 5 + 2^5 - 1 = 41$
	4	$2 \cdot 9 + 2^4 - 1 = \mathbf{33}$
	5	$2 \cdot 13 + 2^3 - 1 = \mathbf{33}$
	6	$2 \cdot 17 + 2^2 - 1 = 37$
	7	$2 \cdot 25 + 2^1 - 1 = 51$

Таким образом, согласно вышеприведённым вычислениям, есть несколько способов переместить восемь дисков за 33 хода. В частности, для случая с восемью кольцами всегда можно использовать $k = \frac{n}{2}$ на каждом шаге алгоритма.

Комментарии. Главная стратегия алгоритма, являющаяся основой этого решения, — «уменьшай и властвуй». Алгоритм явным образом ищет оптимальную величину этого уменьшения на каждом шаге.

Расширить головоломку о Ханойской башне дополнительными стержнями сверх имевшихся трёх в 1889 году предложил французский математик Эдуард Люка́, который придумал оригинальную версию головоломки за несколько лет до этого. Под названием «Головоломка Реве» задача появилась в первом сборнике головоломок Генри Дьюдени «Кентерберийские головоломки» [Dud02], где он приводит решение для $n = 8, 10$ и 21. Впоследствии более детальный анализ вышеприведённого алгоритма позволил получить альтернативные формулы для оптимального значения параметра разбиения k: например, согласно Тэду Роту [Sch80, p. 26–29],

$$k = n - 1 - m, \quad \text{где } m = \left\lfloor \frac{\sqrt{8n-7}-1}{2} \right\rfloor,$$

$$R(n) = \left[n - 1 - \frac{m(m-1)}{2}\right] 2^m + 1,$$

а М. Рэнд [Ran09] дополнительно упростил это выражение до

$$k = n - \lfloor \sqrt{2n} + 0.5 \rfloor.$$

Обобщение этого алгоритма на произвольное число стержней носит название «алгоритм Фрейма — Стюарта». Предполагают, что он оптимален для любого числа стержней, но эта гипотеза не доказана. Другие упоминания этой задачи см. в аннотированной библиографии Дэвида Сингмастера [Sin10, Section 7.M.2.a].

130. Отравленное вино

Решение. а) Определить отравленную бочку за 30 дней можно следующим образом. Пронумеруем бочки от 0 до 999 и представим эти числа как 10-битовые строки, добавив при необходимости к их двоичным разложениям нули в начале. Например, бочки 0 и 999 будут представлены строками 0000000000 и 1111100111 соответственно. Пусть первый раб отопьёт из всех бочек, самый правый знак в разложении которых есть 1, второй раб — из всех бочек, для которых 1 стоит на второй позиции справа, и т. д. — десятому рабу придётся отпить из всех бочек, самый левый знак в разложении которых есть 1. Заметим, что рабам можно позволить выпить такой «коктейль» из многих бочек, поскольку яд, даже разбавленный вином из хороших бочек, не теряет своей силы. Через 30 дней двоичное представление отравленной бочки можно будет определить следующим образом: если i-й раб ($1 \leqslant i \leqslant 10$) — тот, который «отвечал» за i-й знак справа — умрёт от яда, то i-й знак в двоичном представлении отравленной бочки есть 1; в противном случае это 0. Например, если через 30 дней от яда умрут только рабы 1, 3 и 10, то номер отравленной бочки есть $2^0 + 2^2 + 2^9 = 517$.

б) Рассмотрим общий случай задачи: пусть у короля есть N бочек, одна из которых отравлена, и D «тестовых» дней. (В нашей головоломке $N = 1000$ и $D = 5$, поскольку до наступления пятинедельного срока рабы могут умирать на 30, 31, 32, 33 и 34-й день.) Предположим, что король пожертвовал S рабами. Тогда число возможных исходов равняется $(D+1)^S$, поскольку каждый из рабов либо выживет, либо умрёт в один из D дней. Таким образом, имеем

$$(D+1)^S \geqslant N, \quad \text{или} \quad S \geqslant \lceil \log_{D+1} N \rceil .$$

Это неравенство даёт нам нижнюю границу для оптимального числа рабов.

Чтобы использовать именно это число рабов, сделаем следующее. Пронумеруем бочки от 0 до $N-1$ и представим эти числа в системе счисления с основанием $D+1$. Возьмём столько рабов, сколько цифр в самом большом номере бочки. Раб номер k будет «отвечать» за цифру номер k в номере бочки. Пусть k-я цифра в номере бочки есть d. Если $d = 0$, то k-й раб не пьёт из этой бочки; в противном случае он пьёт из неё в день d. Результат будет кодировать отравленную бочку.

Для чисел, данных в условии головоломки, $\lceil \log_{5+1} 1000 \rceil = 4$.

Комментарии. Определение числа через определение знаков в его двоичном представлении — очень старая идея, ей по меньшей мере 500 лет (см. [Sin10, Section 7.M.4]). Она тесно связана с *двоичным поиском* (см., например, [Lev06, разд. 4.3]), с тем преимуществом, что итерации можно производить параллельно. В части б) используется эта идея параллельности.

Различные версии этой головоломки появлялись в колонке Мартина Гарднера в *Scientific American* в ноябре 1965 года (вторично опубликована в [Gar06, задача 9.23]) и в книге [Sha02, p. 16–22]. Головоломка также активно обсуждалась в интернете, на сайтах, посвящённых задачам, которые предлагаются на собеседованиях.

Вышеизложенное решение для части б) авторам прислал Леонид Брагинский.

131. Задача о шашках Тэта

Решение. Ниже приведено решение для $n = 3$. Это единственный случай задачи, в решении которого используется четыре пустых места слева от данного ряда; во всех остальных случаях используется только два.

				1	2	3	4	5	6
				B	W	B	W	B	W
		W	B	B			W	B	W
		W	B	B	B	W	W		
W	W	W	B	B	B				

Теперь решение для $n = 4$. Обратите внимание на ситуацию WBBW_ _BBWW, после которой последние два хода становятся очевидными:

		1	2	3	4	5	6	7	8
		B	W	B	W	B	W	B	W
W	B	B	W	B	W	B			W
W	B	B	W			B	B	W	W
W			W	B	B	B	B	W	W
W	W	W	W	B	B	B	B		

Вот решение для $n = 5$:

		1	2	3	4	5	6	7	8	9	10
		B	W	B	W	B	W	B	W	B	W
W	B	B	W	B	W	B	W	B			W
W	B	B	W			B	W	B	B	W	W
W	B	B	W	W	B	B			B	W	W
W			W	W	B	B	B	B	B	W	W
W	W	W	W	W	B	B	B	B	B		

Решение для $n = 6$:

		1	2	3	4	5	6	7	8	9	10	11	12
		B	W	B	W	B	W	B	W	B	W	B	W
W	B	B	W	B	W	B	W	B	W	B			W
W	B	B	W	B	W	B	W			B	B	W	W
W	B	B			W	B	W	W	B	B	B	W	W
W	B	B	W	W	W	B			B	B	B	W	W
W			W	W	W	B	B	B	B	B	B	W	W
W	W	W	W	W	W	B	B	B	B	B	B		

Решение для $n = 7$:

		1	2	3	4	5	6	7	8	9	10	11	12	13	14
		B	W	B	W	B	W	B	W	B	W	B	W	B	W
W	B	B	W	B	W	B	W	B	W	B	W	B			W
W	B	B	W	B	W			B	W	B	W	B	B	W	W
W	B	B	W	B	W	W	B	B			W	B	B	W	W
W	B	B	W			W	B	B	B	W	W	B	B	W	W
W	B	B	W	W	W	W	B	B	B			B	B	W	W
W			W	W	W	W	B	B	B	B	B	B	B	W	W
W	W	W	W	W	W	W	B	B	B	B	B	B	B		

Начиная с $n \geqslant 8$, можно получать решение рекурсивно, сводя задачу к случаю $n - 4$. В результате первых двух ходов ряд преобразуется в WBBW, затем два пробела, затем $2n - 8$ шашек чередующихся цветов и затем BBWW:

		1	2	3	4	5	6	7	8		$2n-5$	$2n-4$	$2n-3$	$2n-2$	$2n-1$	$2n$
		B	W	B	W	B	W	B	W	…	B	W	B	W	B	W
W	B	B	W	B	W	B	W	B	W	…	B	W	B			W
W	B	B	W			B	W	B	W	…	B	W	B	B	W	W

Последовательность $2n - 8$ чередующихся шашек с двумя пробелами слева приводит к той же задаче, но уже сложности $n - 4 \geqslant 4$. После того как мы решим её рекурсивно, останется сделать всего два хода, чтобы получить решение для ряда из n шашек:

		1	2	3	4	5	6			$2n-5$	$2n-4$	$2n-3$	$2n-2$	$2n-1$	$2n$
W	B	B	W	W	…	W	B	…	B			B	B	W	W
W			W	W	…	W	B	…	B	B	B	B	B	W	W
W	W	W	W	W	…	W	B	…	B	B	B	B	B		

Вышеописанный алгоритм позволяет решить задачу для $2n$ шашек за n ходов. Это утверждение легко доказать, воспользовавшись методом математической индукции или решив рекуррентное соотношение для числа ходов: $M(n) = M(n-4) + 4$ при $n > 7$, $M(n) = n$ при $3 \leqslant n \leqslant 7$.

Комментарии. Алгоритм, представленный выше, основан на стратегии «уменьшай (на 4) и властвуй».

Пространный список публикаций, в которых обсуждается эта или похожая задача, можно найти в библиографии Д. Сингмастера [Sin10, Section 5.O]. Самую первую статью опубликовал Питер Гатри Тэт в 1884 году; несколько авторов утверждают, что решение задачи в общем случае принадлежит Анри Деланнуа, французскому военному офицеру и математику.

132. Солдаты Конвея

Решение. а) Чтобы продвинуть один из колышков на три ряда за линию, естественно стремиться к известной конфигурации, из которой можно продвинуть один из колышков на два ряда за линию, только сдвинутой на один ряд вверх (рис. 4.88, *а*). Соответствующее положение 8 колышков показано на рис. 4.88, *б*. Ещё один способ расположить 8 колышков так, чтобы 1 из них можно было продвинуть на три ряда за линию, показан на рис. 4.88, *в*.

б) Здесь логично воспользоваться решением части а), сдвинутым на один ряд за линию, а затем искать начальную конфигурацию 20 колышков под линией, из которой можно к нему прийти. На рис. 4.89 показано, как привести этот план в действие.

Рисунок 4.89, *а* показывает конфигурацию, полученную в части а), из которой можно достичь клетки на три ряда выше, обозначенной через X. Эти 8 колышков разбиты на пять групп — клетки одной группы обозначены номерами от 1 до 5. На рис. 4.89, *б* представлена комбинация из 20 колышков, которую можно преобразовать в комбинацию на рис. 4.89, *а* и решить задачу. Позиция на рис. 4.89, *б* тоже разбита на пять групп, и клетки соответствующих групп

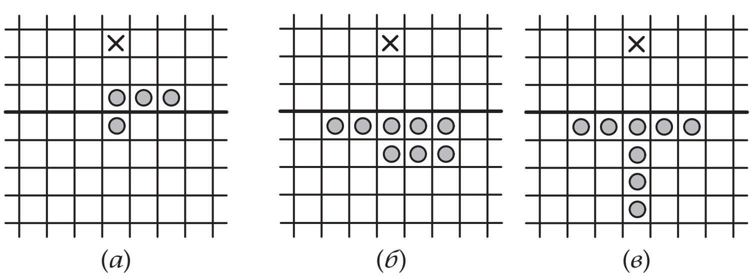


Рис. 4.88. (*а*) Промежуточная искомая конфигурация (через X обозначена клетка-цель); (*б*) и (*в*): показана начальная конфигурация 8 колышков, из которой можно достичь цели

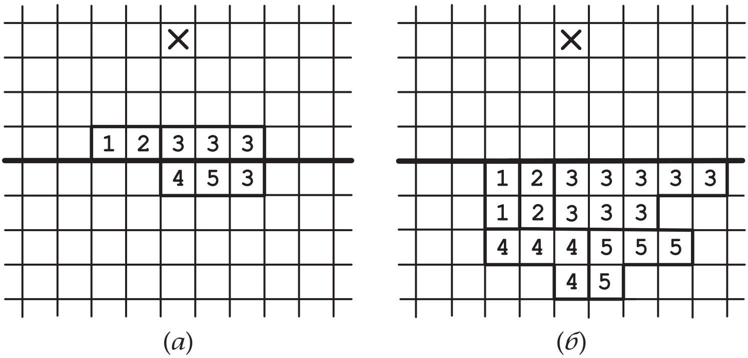


Рис. 4.89. (*а*) Промежуточная искомая конфигурация из 8 колышков. (*б*) Начальная конфигурация 20 колышков, из которой можно достичь цели

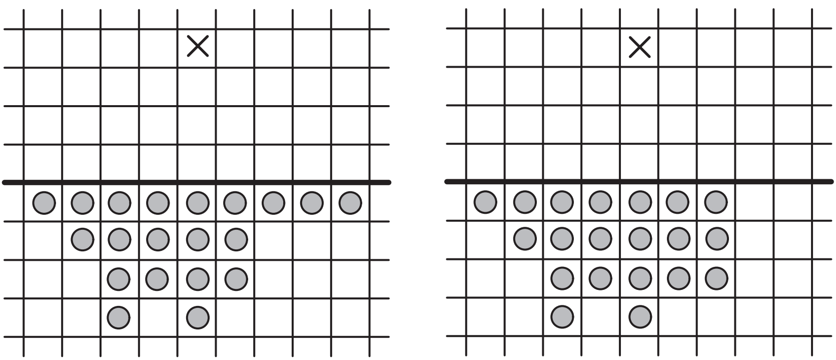


Рис. 4.90. Два альтернативных способа расположить 20 колышков так, чтобы можно было достичь четвёртого ряда за линией

обозначены теми же номерами. Пары колышков, обозначенных через 1 и 2 на рис. 4.89, *б*, становятся одним колышком с тем же обозначением на рис. 4.89, *а*; 8 колышков с номером 3 на рис. 4.89, *б* становятся 4 колышками с тем же номером 3 на рис. 4.89, *а*, и т. д.

Расположить 20 колышков так, как показано на рис. 4.89, *б*, — не единственный способ решить задачу; в книге [Bea92, p. 212] описаны два других, показанных на рис. 4.90.

Комментарии. В принципе, вышеизложенное решение основано на стратегии «преобразуй и властвуй», но помогает и мышление в духе стратегии «разделяй и властвуй».

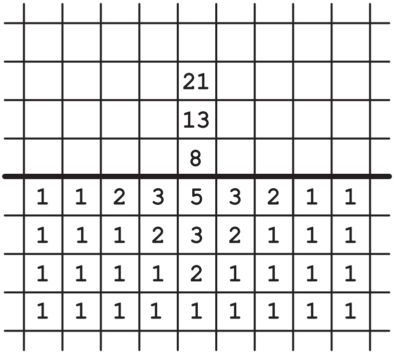


Рис. 4.91. Функция пагоды, которая позволяет доказать, что необходимо минимум 8 колышков, чтобы достичь клетки в третьем ряду

Мы не просим читателя доказать, что данное в задаче число колышков — наименьшее, которое необходимо, чтобы достичь указанных рядов. Чтобы это доказать, нужно использовать специальную функцию «*resource counts*» (второе название *функция пагоды*), которую придумали Дж. Х. Конвей и Дж. М. Бордман в 1961 году [Bea92, p. 71]. Функция пагоды присваивает каждой клетке доски число так, чтобы после любого допустимого хода-прыжка сумма присвоенных занятым колышками клеткам чисел была не больше, чем до прыжка. Среди бесконечного множества возможных функций пагоды, Дж. Д. Бисли в [Bea92, p. 212] выбирает такую (см. рис. 4.91), с использованием которой можно доказать, что, имея менее 8 колышков изначально, нельзя достичь клетки-цели в третьем ряду над линией. (Предполагается, что все значения, отсутствующие на рисунке, есть 1 для клеток под линией и получаются сложением двух нижних значений для клеток над линией. Значение 21 присвоено клетке в третьем ряду над линией — клетке-цели.) Поскольку для любого набора из семи или менее клеток под линией функция пагоды не превышает $5\cdot 1+3\cdot 3+2\cdot 3 = 20$, то семь или менее колышков ни с каких клеток под линией не могли бы дойти до клетки с функцией пагоды, равной 21.

Возможно, наиболее впечатляющее приложение функции пагоды придумал сам Дж. Х. Конвей — в 1961 году он доказал, что никакая группа колышков не смогла бы достичь пятого ряда над линией. Чтобы доказать этот противоречащий интуиции факт, Конвей использовал степени числа $\frac{\sqrt{5}-1}{2}$, обратного к так называемому *золотому сечению* $\frac{\sqrt{5}+1}{2}$. Подробности этого замечательного доказательства можно найти в классической книге о математических играх [Ber04], в которой Конвей был соавтором, а также на нескольких сайтах (заметим, что головоломку также называют «Послать разведчиков в пустыню».) Дж. Тэнтон доказал тот же факт, используя функцию пагоды, основанную на числах Фибоначчи [Tan01, p. 197–198].

133. Игра «Жизнь»

Решение. Самые маленькие натюрморты («блок» и «бочка»), осциллятор («семафор») и космический корабль («планёр») показаны на рис. 4.92.

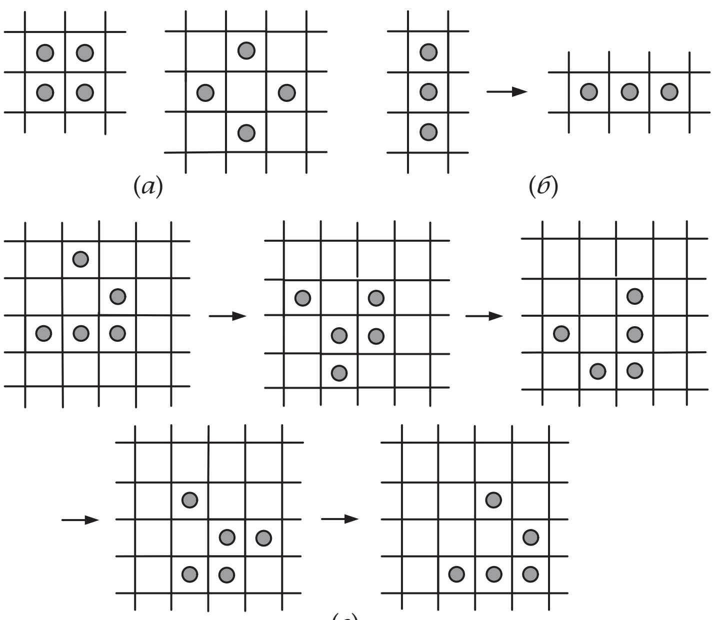


Рис. 4.92. Конфигурации в игре «Жизнь»: (*а*) «блок» и «бочка»; (*б*) «семафор»; (*в*) «планёр», за четыре поколения сдвигающийся на одну клетку по диагонали вниз и направо

Комментарии. В этой головоломке нужно определить, какие входные данные приводят к некоторому особому результату при применении алгоритма.

Игру «Жизнь» придумал британский математик Джон Конвей в 1970 году. Она стала широко известной после того, как Мартин Гарднер написал о ней в своей колонке в *Scientific American* [Gar83, гл. 20–22]. Она до сих пор вызывает неослабевающий интерес, судя по множеству посвящённых ей сайтов. Интересна игра по нескольким причинам. Во-первых, простые правила Конвея приводят к удивительным и довольно неожиданным схемам развития игры. Во-вторых, игру можно использовать как модель универсального компьютера [Ber04, Chapter 25], что позволяет связать игру с самыми глубокими вопросами — от механизмов эволюции до природы Вселенной.

134. Раскраска точек

Решение. Задачу можно решить с помощью следующего рекурсивного алгоритма. Если $n = 1$, покрасим точку в любой цвет, например в чёрный. Если $n > 1$, действуем следующим образом. Выберем прямую l (горизонтальную или вертикальную), на которой лежит нечётное число точек, подлежащих окраске; если такой прямой нет, выберем любую прямую, на которой лежит хотя бы одна точка. Выберем точку P на прямой l. Рекурсивно покрасим все точки, кроме P, требуемым образом. Мы покажем, что P всегда можно покрасить так, чтобы удовлетворить условиям задачи. Пусть m — другая прямая сетки, проходящая через P. Если число раскрашенных точек на обеих прямых l и m чётно, то ровно половина из них окрашена в чёрный и ровно половина — в белый цвет; поэтому P можно покрасить в любой цвет. Если число окрашенных точек чётно на одной прямой и нечётно на другой, P следует покрасить в такой цвет, чтобы количества разноцветных точек на этой другой прямой стали равными. Если число раскрашенных точек и на прямой l и на прямой m нечётно и среди точек на обеих прямых в один из цветов окрашено на одну точку больше, чем в другой, P следует покрасить так, чтобы количества разноцветных точек на обеих линиях стали равными.

Наконец, нам нужно показать, что если число окрашенных точек и на прямой l, и на прямой m нечётно, то невозможна ситуация, при которой один из цветов — скажем, чёрный — встречается чаще на l, в то время как другой цвет — белый — встречается чаще на m. Действительно, в таком случае общее число точек на l было бы чётно (нечётное число окрашенных точек плюс P). Согласно тому, как выбрана l, отсюда должно следовать, что на всех прямых чётное число точек, и на всех прямых, кроме l и m, ровно половина точек окрашена чёрным и ровно половина белым. Но если мы теперь подсчитаем общее число точек, окрашенных каждым из цветов, сложив число окрашенных точек на всех параллельных l прямых и на самой l, то нам придётся признать, что общее число чёрных точек на один превышает общее число белых; если, с другой стороны, подсчитать общее число точек, окрашенных каждым из цветов, сложив число окрашенных точек на всех параллельных m прямых и на самой m, то выйдет, что общее число белых точек на один превышает общее число чёрных. Это противоречие завершает доказательство корректности алгоритма.

Комментарии. Эта задача, которая предлагалась участникам XXVII Международной математической олимпиады, появилась в декабре 1986 года в журнале «Квант» (задача M1019, с. 26). Приведённое решение было предложено А. П. Савиным; оно было опубликовано в апрельском номере 1987 года (с. 26–27). Если пользоваться нашей терминологией, оно основано на варианте стратегии «уменьшай и властвуй», а именно «уменьшай на один».

135. Разные пары

Решение. Ниже мы рассмотрим один из оптимальных способов сформировать $2n-1$ наборов разных пар. Для удобства пронумеруем детей от 1 до $2n$ и поместим эти числа в таблице $2 \times n$. Пары для первого набора получаются из столбцов этой таблицы. Чтобы получить остальные $2n-2$ наборов, будем вращать в таблице, скажем по часовой стрелке, все числа, кроме 1. На рис. 4.93 показан пример для $n = 3$.

1	2	3
6	5	4

1	6	2
5	4	3

1	5	6
4	3	2

1	4	5
3	2	6

1	3	4
2	6	5

Рис. 4.93. Пять наборов по три разные пары

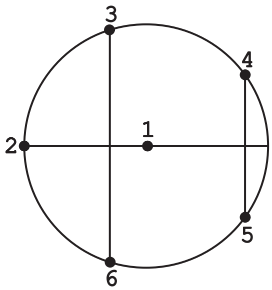


Рис. 4.94. Геометрический способ формировать пары для $n = 3$

Другой способ описать алгоритм: поместим число 1 в центр круга, а на его окружности расположим $2n - 1$ точек на равном расстоянии друг от друга и пронумеруем их по часовой стрелке от 2 до $2n$. Набор пар получится, если провести сначала диаметр через центр и одну из точек окружности (так получится одна пара), а затем хорды, перпендикулярные этому диаметру (так получатся остальные $n - 1$ пар). Рисунок 4.94 иллюстрирует этот подход при $n = 3$. Новые наборы пар формируются при вращении диаметра — тогда и у центральной точки появится новая пара, и остальные точки соединятся в пары по-новому, после того как мы проведём новые хорды.

Комментарии. Можно считать, что этот алгоритм основан на стратегии замены представления.

Обе вышеописанные трактовки алгоритма были даны Морисом Крайчиком в «Математических развлечениях» [Kra53, p. 226–227]. Конечно, эта задача — переформулированная задача планирования игр для кругового турнира с $2n$ участниками.

136. Поимка шпиона

Решение. Поскольку шпион находится в позиции a в момент $t = 0$ и перемещается на b единиц за каждый временной интервал, позиция шпиона в момент t может быть вычислена по очевидной формуле $x_{ab}(t) = a + bt$. Поэтому для решения задачи достаточно придумать алгоритм, который сгенерирует набор всех целых пар (a, b) в виде последовательности и в моменты $t = 0, 1\ldots$ будет проверять ту позицию на нахождение в ней шпиона, которая вычислена по этой формуле для соответствующих (a, b) из последовательности. Тогда, какими бы ни были определяющие движение шпиона параметры a и b, алгоритм дойдёт до них после конечного числа шагов. Очевидно, что если в момент $t = 0$ шпион находится в позиции 0, его можно найти, проверив эту позицию в момент $t = 0$, вне зависимости от значения b;

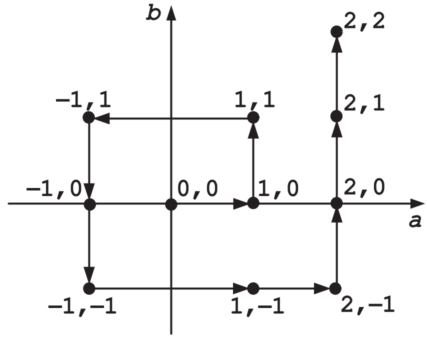


Рис. 4.95. Решение головоломки «*Поимка шпиона*» с помощью спирали

следовательно, после того как мы проверим пару $(0,0)$ в момент $t = 0$, никаких других пар вида $(0,b)$ проверять будет не нужно.

Целые пары (a,b) можно пронумеровать множеством различных способов. В частности, эти пары можно представить как целочисленные точки на декартовой плоскости и пронумеровать их, двигаясь по спирали из $(0,0)$ (рис. 4.95). Тогда проверять местонахождение шпиона нужно будет в следующем порядке:

$$0 + 0 \cdot 0 = 0, \quad 1 + 0 \cdot 1 = 1, \quad 1 + 1 \cdot 2 = 3, -1 + 1 \cdot 3 = 2,$$
$$-1 + 0 \cdot 4 = -1, \quad -1 - 1 \cdot 5 = -6, \quad 1 - 1 \cdot 6 = -5, \ldots$$

Другой способ пронумеровать целочисленные точки на плоскости — воспользоваться стандартным методом из математической теории множеств. При использовании этого метода пары (a,b) считаются элементами бесконечной матрицы X, строки и столбцы которой соответствуют различным значениям a $(a = 0, \pm 1 \pm 2, \ldots)$ и b $(b = 0, \pm 1, \pm 2, \ldots)$ (рис. 4.96). Затем элементы матрицы нумеруются по её диагоналям, которые идут с «северо-востока» на «юго-запад», как показано на рис. 4.96.

Последовательность позиций, которые мы будем проверять, если решим использовать этот алгоритм, будет начинаться следующим образом:

$$0 + 0 \cdot 0 = 0, \quad 1 + 0 \cdot 1 = 1, \quad 1 + 1 \cdot 2 = 3, \quad -1 + 0 \cdot 3 = -1,$$
$$1 + (-1) \cdot 4 = -3, \quad -1 + 1 \cdot 5 = 4, \quad 2 + 0 \cdot 6 = 2.$$

Комментарии. Первое из вышеприведённых решений предложил авторам Стивен Лукас из Университета Джеймса Медисона, один из рецензентов этой книги. Кажется, оно проще, чем решение самих авторов, которое основано на

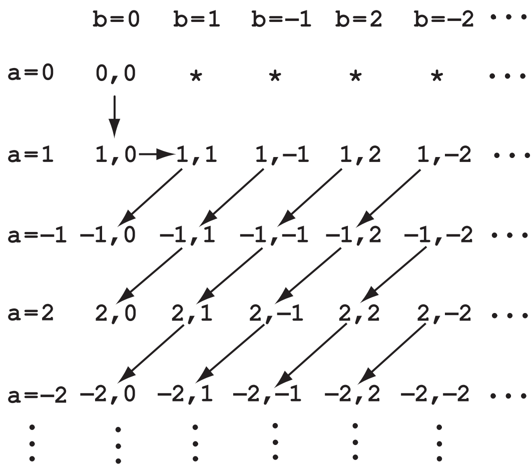


Рис. 4.96. Решение головоломки «*Поимка шпиона*» с помощью матрицы. Элементы нулевой строки, за исключением x_{00}, обозначены звёздочками, поскольку в алгоритме не используются

нумерации по диагоналям бесконечной матрицы. Оба решения, пользуясь нашей терминологией, являют собой пример замены представления.

Авторы наткнулись на эту задачу, читая Интернет-коллекцию головоломок [Leino]. Мы не смогли определить, где впервые появилась эта головоломка, что не так уж удивительно, если вспомнить, о чём она сама.

137. Прыжки в пары — 2

Решение. У головоломки есть решение тогда и только тогда, когда n делится на 4.

Очевидно, что задачу нельзя решить для нечётного числа монет, поскольку в конечном состоянии монеты организованы в пары, т. е. их общее число чётно.

Более того, n должно делиться на 4. Чтобы это доказать, рассмотрим состояние задачи перед последним ходом: $(n-2)$ монет уже в парах, и, следовательно, одна из двух остающихся одиночных монет должна перепрыгнуть через чётное число монет, чтобы оказаться у другой одиночной монеты. Поскольку монета должна перепрыгнуть через чётное число монет на ходе $i, 1 \leqslant i \leqslant \frac{n}{2}$, тогда и только тогда, когда i

чётно, номер последнего хода, $\frac{n}{2}$, должен быть чётным, откуда следует, что n должно делиться на 4.

Алгоритм решения задачи можно придумать, используя обратный проход следующим образом. Рассмотрим конечное состояние головоломки, в котором n монет ($n = 4k$, $k > 0$) объединены в пары, а пары пронумерованы слева направо от 1 до $\frac{n}{2}$. Чтобы прийти к начальному состоянию, в котором у нас есть лишь n одиночных монет, расположенных в ряд, возьмём верхнюю монету из пары с номером $\frac{n}{4}+1$ и переместим её налево через все $\frac{n}{2}$ монет. Затем возьмём верхнюю монету из пары с номером $\frac{n}{4}$ и переместим её налево через все $\frac{n}{2}-1$ монет. Продолжим перемещать верхние монеты из пар через все монеты слева от них до тех пор, пока не передвинем верхнюю монету самой левой пары налево через $\frac{n}{4}$ монет. Затем, начиная с самой левой из оставшихся пар (той, у которой изначально был номер $\frac{n}{4}+2$) и заканчивая самой правой парой (изначальный номер которой был $\frac{n}{2}$), переместим верхнюю монету пары налево через $\frac{n}{4}-1, \ldots, 1$ монет соответственно. Верхние монеты нужно класть как одиночные только что описанным образом — вспомним, что по условию задачи мы игнорируем пустое пространство между соседними монетами.

«Обращая» вышесказанное, мы получим следующий алгоритм формирования пар из n монет, лежащих в ряд и пронумерованных слева направо от 1 до n. Сначала выполним для $i = 1, 2, \ldots, \frac{n}{4}-1$ следующую операцию: для самой правой одиночной монеты найдём монету слева от неё такую, чтобы между ними было i монет, и поместим эту монету на самую правую одиночную. Затем для $i = \frac{n}{4}, \frac{n}{4}+1, \ldots, \frac{n}{2}$ повторим следующую операцию: возьмём самую левую одиночную монету и перенесём её через i монет направо — она окажется на одиночной монете. Шаги алгоритма для $n = 8$ изображены на рис. 4.97. Очевидно, что этот алгоритм делает минимальное возможное число шагов, поскольку на каждом шаге формируется новая пара.

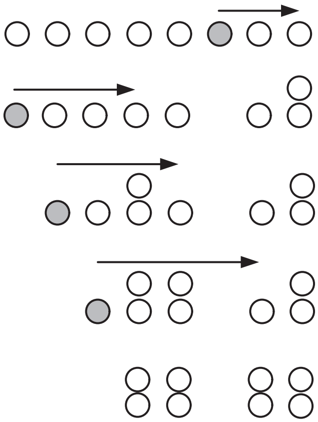


Рис. 4.97. Решение головоломки «*Прыжки в пары — 2*» для $n = 8$. Прыгающие монеты нарисованы серым

Комментарии. Решение этой головоломки — отличный пример того, как порой полезно в разработке алгоритмов использовать обратный проход.

Мартин Гарднер приписывает головоломку и её решение У. Ллойду Миллигану (см. [Gar83, p. 172, 180]).

138. Делёж конфет

Решение. Пусть i и j соответственно — количества конфет у ребёнка и у его/её соседа справа в момент перед свистком. После свистка у того ребёнка, у которого было i конфет, будет i' конфет, где i' определяется следующей формулой:

$$i' = \begin{cases} \frac{i}{2} + \frac{j}{2}, & \text{если это число чётно,} \\ \frac{i}{2} + \frac{j}{2} + 1 & \text{в противном случае.} \end{cases}$$

Из этой формулы следует, что если $i = M$, где M — число конфет у того ребёнка, у которого изначально конфет больше всех (напомним, что оно чётно), то $i' \leqslant M$. Таким образом, ни у какого ребёнка ни в какой момент не может оказаться больше M конфет.

Пусть теперь m — наименьшее число конфет, которыми обладает один из детей в момент перед некоторым свистком. После этого свистка у каждого ребёнка будет минимум $\frac{m}{2} + \frac{m}{2} = m$ конфет; более того, у ребёнка будет не меньше $m + 1$ конфет, если только перед свистком у этого ребёнка и у его соседа справа не было ровно m конфет.

Обобщим эти рассуждения: если у $1 \leqslant k < n$ детей подряд было m конфет, а у $(k+1)$-го ребёнка (считая против часовой стрелки) их было больше m, то у первых $k-1$ из них после свистка будет по-прежнему m конфет, но у k-го ребёнка их будет больше, чем m. Следовательно, после k итераций минимальное число конфет среди детей возрастёт. Поскольку число конфет, которое может быть у ребёнка, ограничено также и сверху, процесс после конечного числа итераций приведёт к тому, что у всех детей будет одинаковое число конфет.

Комментарии. В задаче используется идея полуинварианта — полуинвариантом является наименьшее число конфет, которыми обладает один из детей в момент перед некоторым свистком. (Понятие полуинварианта обсуждается в первой части учебного раздела.)

Эта задача довольно-таки известная; в частности, она предлагалась на Пекинской олимпиаде 1962 года и на Ленинградской городской олимпиаде по математике 1983 года. Эту же задачу в несколько изменённом виде можно найти в статье [Iba03].

139. Круглый Стол короля Артура

Решение. Поскольку у каждого рыцаря не меньше $\frac{n}{2}$ друзей, врагов у него не больше $n-1-\frac{n}{2}=\frac{n}{2}-1$. Если $n=3$, каждый из трёх рыцарей дружит с двумя другими, поэтому они могут сидеть как угодно. Если $n>3$, то можно сначала рассадить рыцарей произвольным образом и подсчитать число пар врагов, сидящих друг рядом с другом. Если это число равно нулю, то цель достигнута; если нет — это число можно уменьшить по крайней мере на один следующим образом. Пусть рыцари A и B являются врагами и сидят рядом, B слева от A (рис. 4.98). Тогда среди друзей рыцаря A существует рыцарь C, сосед которого слева, D, дружит с B. (Если это не так, то у рыцаря B было бы как минимум $\frac{n}{2}$ врагов.) Если мы теперь поменяем местами всех рыцарей между B и C по часовой стрелке, включая самих рыцарей B и C (это показано стрелками на рис. 4.98, *а*), получится, что за столом стало на одну пару врагов-соседей меньше (рис. 4.98, *б*).

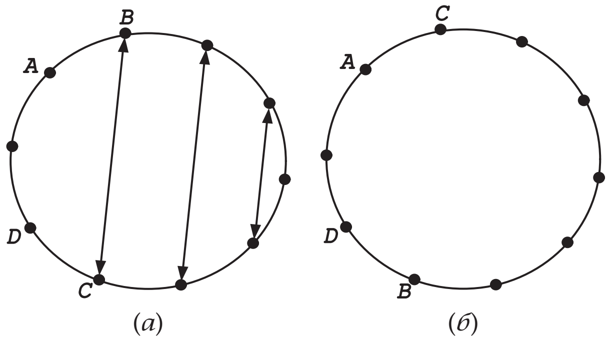


Рис. 4.98. Пошаговое улучшение ситуации за столом в головоломке «*Круглый Стол короля Артура*»

Комментарии. Головоломка и её решение взяты из статьи о полуинвариантах [Kur89], опубликованной в журнале «Квант» в 1989 году. (Мы опустили требование, чтобы число рыцарей было чётным, поскольку оно не является необходимым.) Нужно заметить, что существование решения у головоломки следует из *теоремы Дирака*: в любом графе с $n \geqslant 3$ вершинами, в котором степень каждой вершины (т. е. число вершин, соединённых с ней ребром) равняется хотя бы $\frac{n}{2}$, есть гамильтонов цикл. В этой головоломке вершины графа — это рыцари, а две вершины соединены ребром в том и только в том случае, когда рыцари, соответствующие вершинам, дружат.

140. Снова задача об n ферзях

Решение. В решении для $n = 4$, показанном на рис. 4.99, *а*, можно увидеть следующую структуру решения для других чётных n. Поместим ферзей на строки $2, 4, \ldots, \frac{n}{2}$ первых $\frac{n}{2}$ столбцов и на строки $1, 3, \ldots, n-1$ последних $\frac{n}{2}$ столбцов. Действительно, это будет работать не только для любого $n = 4+6k$, но и для любого $n = 6k$ (пример см. на рис. 4.99, *б*). Более того, поскольку ни в одном из этих решений ферзь не стоит на главной диагонали, из любого решения можно получить решение для на единицу большего n, просто добавив ферзя в последнюю строку последнего столбца (пример см. на рис. 4.99, *в*).

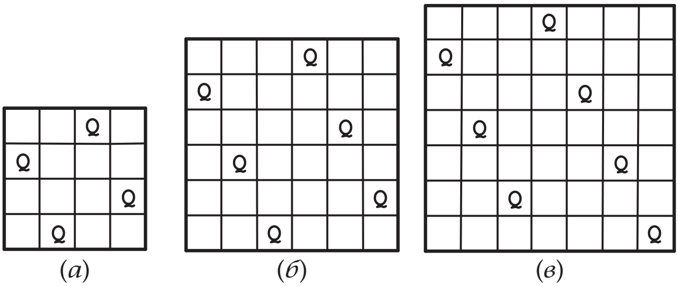


Рис. 4.99. Решение головоломки «*Снова задача об n ферзях*» для (*а*) $n = 4$, (*б*) $n = 6$ и (*в*) $n = 7$

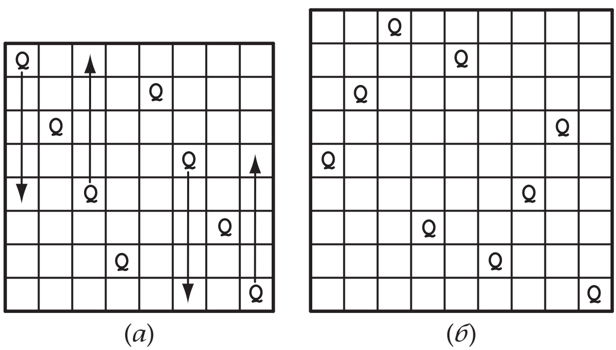


Рис. 4.100. Решение головоломки «*Снова задача об n ферзях*» для (*а*) $n = 8$ и (*б*) $n = 9$

К сожалению, это не работает для $n = 8 + 6k$. Несмотря на то, что на такую доску можно поместить ферзей так же, как это было сделано выше, а затем переставить их, легче начать ставить ферзей на строки с нечётными номерами первых $\frac{n}{2}$ столбцов, а потом — на строки с чётными номерами. Затем можно получить позицию, в которой ферзи не атакуют друг друга, поменяв строками ферзей в столбцах 1 и $\frac{n}{2} - 1$ и ферзей в столбцах $\frac{n}{2} + 2$ и n (см. пример для $n = 8$ на рис. 4.100, *а*).

Поскольку в полученном решении на главной диагонали нет ферзей, из него можно получить решение для $n = 9 + 6k$, поместив ферзя в последнюю строку последнего столбца расширенной доски (рис. 4.100, *б*).

Таким образом, имеем следующий алгоритм, который даёт номера строк, на которые столбец за столбцом следует поместить $n > 3$ ферзей на доске $n \times n$.

Подсчитать остаток r от деления n на 6.

Случай 1 (r не равняется ни 2, ни 3): Записать последовательно чётные числа от 2 до n включительно, а затем добавить в список последовательные нечётные числа от 1 до n включительно.

Случай 2 (r равно 2): Записать последовательно нечётные числа от 1 до n включительно, а затем поменять местами первое и предпоследнее числа. Затем добавить в список последовательные чётные числа от 2 до n включительно и после поменять местами число номер 4 и последнее число получившегося списка.

Случай 3 (r равно 3): Применить указания из Случая 2 к $n-1$, а не к n, и добавить в получившийся список n.

Комментарии. Вышеописанный алгоритм, следующий схеме, приведённой в работе [Gin06], довольно-таки прямолинеен — в том смысле, что ферзи помещаются на первый же доступный квадрат каждого столбца доски. Нетривиально то, что для некоторых n мы должны начинать со второго квадрата первого столбца, а для некоторых n — поменять местами две пары ферзей, чтобы ни один не атаковал другого. Также стоит отметить, что решения для досок $n \times n$, где n нечётно, на самом деле получаются просто при расширении решений для досок $(n-1) \times (n-1)$.

Задача об n ферзях — одна из самых известных задач занимательной математики, впервые она привлекла к себе внимание математиков в середине XIX века. Поиски эффективного алгоритма для её решения начались, конечно, гораздо позже. Среди множества подходов стандартным стало решение задачи путём поиска с возвратом — как в первой части учебного раздела. Помимо образовательной ценности, применение поиска с возвратом (по крайней мере, в принципе) может помочь найти все решения задачи. Если же требуется найти всего одно решение, то можно использовать и более простые методы. Обзорная статья [Bel09] содержит больше полудюжины наборов формул для прямого вычисления позиций ферзей, включая самый ранний, придуманный Е. Паулсом и опубликованный в 1874 г. Как ни странно, его подход полностью решал задачу, но не был замечен вплоть до того момента, когда Б. Бернхардссон напомнил специалистам о его существовании [Ber91].

141. Задача Иосифа Флавия

Решение. Пусть $J(n)$ — номер выжившего. Удобно отдельно рассмотреть случаи чётного и нечётного n. Если n чётно, т. е. $n = 2k$, первый обход круга приведёт к той же задаче, но для вдвое меньшего числа людей. Единственная разница заключается в нумерации; например, человек, изначально стоявший на позиции 3, будет на позиции 2 при втором проходе, человек, изначально стоявший на позиции 5, будет на позиции 3, и т. д. Таким образом, чтобы найти начальную позицию какого-либо человека, нужно просто умножить его новую позицию на два и вычесть один. В частности, для выжившего получаем

$$J(2k+1) = 2J(k) - 1.$$

Теперь рассмотрим случай нечётного $n > 1$, т. е. $n = 2k + 1$. При первом проходе погибнут люди на всех чётных позициях. Сразу же после этого погибнет человек на позиции 1, после чего получится та же задача, но для k человек. Чтобы теперь получить из новой нумерации старую, нужно умножить новую позицию на 2 и добавить 1. Таким образом, для нечётных значений n мы имеем

$$J(2k+1) = 2J(k) + 1.$$

Чтобы найти явную формулу для $J(n)$, последуем схеме, обрисованной в [Gra94, разд. 1.3]. Используя начальное условие $J(1) = 1$ и приведённые выше рекуррентные соотношения для чётных и нечётных значений n, мы получим следующие значения $J(n)$ для $n = 1, 2, \ldots, 15$:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$J(n)$	1	1	3	1	3	5	7	1	3	5	7	9	11	13	15

Приглядевшись к этим значениям, нетрудно заметить, что для значений n, лежащих между двумя ближайшими степенями 2, т. е. для $2^p \leqslant n < 2^{p+1}$, иначе говоря, для $n = 2^p + i$, где $i = 0, 1, \ldots, 2^p - 1$, соответствующие значения $J(n)$ пробегают все нечётные числа от 1 до $2^{p+1} - 1$. Это наблюдение можно выразить формулой

$$J(2^p + i) = 2i + 1 \quad \text{при } i = 0, 1, \ldots, 2^p - 1.$$

Несложно с помощью индукции по p доказать, что эта формула и есть решение рекуррентных соотношений для задачи Иосифа Флавия при любом неотрицательном целом p. Для базы индукции, $p = 0$, имеем $J\left(2^0 + 0\right) = 2 \cdot 0 + 1 = 1$, как и должно быть согласно начальному условию. Теперь, предполагая, что для данного неотрицательного целого p и любого $i = 0, 1, \ldots, 2^p - 1$ выполнено соотношение $J\left(2^p + i\right) = 2i + 1$, покажем, что

$$J(2^{p+1} + i) = 2i + 1 \quad \text{при } i = 0, 1, \ldots, 2^{p+1} - 1.$$

Если i чётно, то можно представить его как $2j$, где $0 \leqslant j < 2^p$. Затем, используя предположение индукции, получаем

$$\begin{aligned} J(2^{p+1} + i) = J(2^{p+1} + 2j) = J(2(2^p + j)) = 2J(2^p + j) - 1 = \\ = 2(2j + 1) - 1 = 2i + 1. \end{aligned}$$

Если i нечётно, то можно представить его как $2j + 1$, где $0 \leqslant j < 2^p$. Затем, используя предположение индукции, получаем

$$\begin{aligned} J(2^{p+1} + i) = J(2^{p+1} + 2j + 1) = J(2(2^p + j) + 1) = 2J(2^p + j) + 1 = \\ = 2(2j + 1) + 1 = 2i + 1. \end{aligned}$$

Отметим также, что $J(n)$ можно получить и с помощью циклического сдвига двоичного представления числа n на один бит влево [Gra94, p . 12]. Например, если $n = 40 = 101000_2$, то $J\left(101000_2\right) = 10001_2 = 17$.

И наконец, можно найти $J(n)$ и с помощью формулы

$$J(n) = 1 + 2n - 2^{1 + \lfloor \log_2 n \rfloor},$$

упомянутой в [Weiss]. Если взять для примера то же самое значение 40, получим $J\left(40\right) = 1 + 2 \cdot 40 - 2^{1 + \left\lfloor \log_2 40 \right\rfloor} = 17$.

Комментарии. Эту головоломку назвали в честь Иосифа Флавия, еврейского историка, который участвовал в еврейском восстании против римлян 66–70 гг. и описал его. Иосиф, будучи военачальником, удерживал крепость Йотапата 47 дней, а после того как город пал, укрылся вместе с 40 сторонниками в пещере неподалёку. Там восставшие решили скорее умереть, чем сдаться. Иосиф предложил встать в круг и, проходя по этому кругу, убивать каждого третьего до тех пор, пока не останется только один человек, который должен будет убить себя сам. Иосиф придумал, как встать в круг так, чтобы выжить, и когда остались в живых только он и ещё один человек, Иосиф уговорил свою несостоявшуюся жертву сдаться римлянам.

В нашей головоломке убивают каждого *второго*, из-за чего решить её легче. Другие варианты и исторические ссылки см. в аннотированной библиографии Дэвида Сингмастера [Sin10, Section 7.B]; среди множества других не столь исчерпывающих источников стоит упомянуть книгу [Bal87, p. 32–36].

142. Двенадцать монет

Решение. Дерево решений на рис. 4.101 даёт алгоритм для решения задачи о фальшивой монете, если всего монет 12, за три взвешивания. В этом дереве монеты пронумерованы от 1 до 12. Внутренние вершины обозначают взвешивания, причём взвешиваемые монеты перечислены внутри вершины. Например, корень соответствует первому взвешиванию, при котором монеты 1, 2, 3, 4 и монеты 5, 6, 7, 8 кладутся на левую и правую чаши весов соответственно. Рёбра, ведущие от вершины к детям, обозначены в соответствии с результатом взвешивания, описанного внутри вершины: «<» значит, что левая чаша весов легче правой, «=» значит, что чаши весят одинаково, а «>» значит, что левая чаша весов тяжелее правой. Листья обозначают вывод: «=» означает, что все монеты настоящие, а номер после «+» или «−» означает, что монета с этим номером тяжелее или легче соответственно. Список над внутренней вершиной обозначает, какие выводы всё ещё возможны, прежде чем мы осуществим описанное внутри вершины взвешивание. Например, перед первым взвешиванием возможно как то, что все монеты настоящие, так и то, что некоторая монета либо тяжелее, либо легче остальных.

Задачу нельзя решить меньше чем за три взвешивания. Дерево решений, соответствующее любому алгоритму решения задачи, должно иметь хотя бы $2\cdot 12+1=25$ листьев, отражающих все возможные ответы. Поэтому его высота, равная числу взвешиваний W в худшем случае, должна удовлетворять неравенству $W \geqslant \lceil \log_3 25 \rceil$, или $W \geqslant 3$.

Комментарии. Привлекательность вышеприведённого решения заключается в его симметричности: вторые взвешивания производятся с одними и теми же монетами, в какую бы сторону ни наклонились весы, и в следующих взвешиваниях фигурируют одни и те же пары монет. На самом деле, у задачи есть и совершенно неадаптивное решение, т. е. такое, при котором

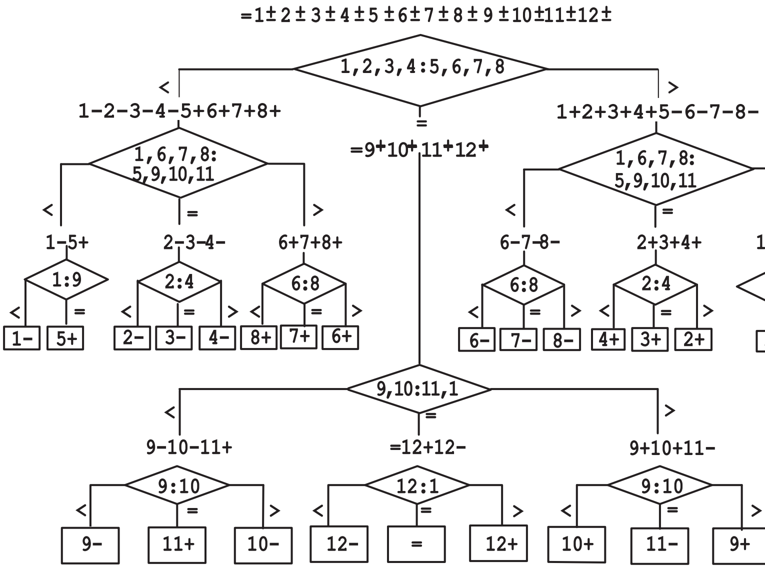


Рис. 4.101. Дерево решений для головоломки «*Двенадцать монет*»

выбор монет для второго взвешивания не зависит от результата первого взвешивания, а выбор монет для третьего взвешивания не зависит ни от исхода первого взвешивания, ни от исхода второго (см., например, [OBe65, p. 22–25]). Другие решения, в том числе одно, основанное на использовании троичной системы, см. в ссылках на интернет-страничке А. Богомольного [Bogom].

Оптимальный алгоритм для общего случая этой задачи при $n \geqslant 3$ монет требует совершить $\lceil \log_3(2n+3) \rceil$ взвешиваний. Этот факт был установлен независимо несколькими математиками, которые опубликовали свои результаты в *Mathematical Gazette* в 1946 г.

Первое упоминание этой знаменитой головоломки в печати относится к 1945 г. [Sin10, Section 5.C]. Похоже, что она появилась либо незадолго до, либо во время Второй мировой войны. Т. Х. О'Бейрн пишет в своей книге, что, как предполагают, эта головоломка «отвлекала на себя столько усилий учёных в военное время, что всерьёз рассматривалось предложение забросить её на вражескую территорию с целью нанесения ущерба» [OBe65, p. 20]. С тех пор эта головоломка стала одной из самых популярных и часто встречается как в сборниках головоломок, так и на посвящённых головоломкам сайтах.

143. Заражённая шахматная доска

Решение. Ответом является число n.

Есть множество возможностей начального расположения n заражённых квадратов, которые в конечном итоге заразят всю доску $n \times n$; на рис. 4.102 показаны две из них.

Чтобы доказать, что n есть наименьшее необходимое для этого число, заметим, что суммарный периметр заражённой области (который, в общем случае, равняется сумме периметров связных подобластей, заражённых вирусом) не увеличивается по мере распространения вируса по доске. Действительно, в момент, когда заражается новый квадрат, по меньшей мере две его стороны оказываются внутри заражённой области и максимум две стороны к ней добавляются. Поэтому, если изначально заражено меньше чем n квадратов, то изначальный общий периметр заражённой области будет меньше чем $4n$, и меньшим чем $4n$ он и останется. Следовательно, вся доска, периметр которой равен $4n$, никогда не будет заражена полностью.

Комментарии. В головоломке используется идея полуинварианта, которую мы упоминали при обсуждении метода итерационного улучшения в первой части учебного раздела, но здесь она используется для доказательства невозможности определённого варианта развития событий — а именно такого, при котором изначально заражённые квадраты в количестве, меньшем n, могли бы заразить всю доску $n \times n$.

Головоломка была включена в сборник [Win04, p. 79]. В книге [Bol07, p. 171] рассмотрена версия этой головоломки,

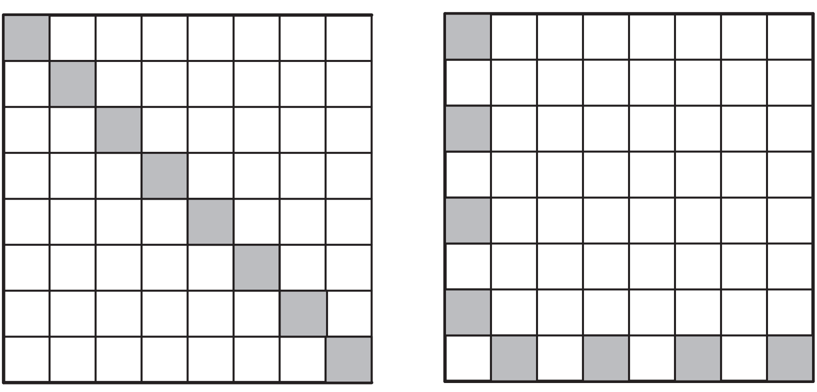


Рис. 4.102. Два варианта начального расположения заражённых квадратов, которые распространятся на всю доску

в которой новая клетка заражается только в том случае, когда у неё есть по меньшей мере три заражённых клетки по соседству.

144. Разрушение квадратов

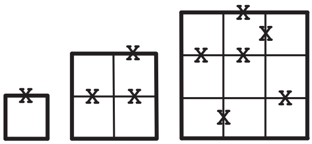


Рис. 4.103. «*Разрушение квадратов*» при $n = 1$, $n = 2$ и $n = 3$

Решение. Нижеприведённый рекурсивный алгоритм позволяет решить задачу, удалив наименьшее число спичек, которое равняется $\left\lceil \frac{n^2}{2} \right\rceil + 1$ при $n > 1$ (и, разумеется, равняется 1 при $n = 1$). Для $n = 1, 2$ или 3 решение приведено на рис. 4.103.

Если $n > 3$, сделаем следующее. Рассмотрим рамку шириной 1, образованную периметром данного квадрата и периметром квадрата размера $n - 2$ внутри него. Начиная с левого верхнего угла рамки и двигаясь против часовой стрелки, будем удалять все спички, которые оказались бы на средней линии костяшки домино при покрытии рамки кольцом из домино, кроме спичек на средней линии последней костяшки в кольце. Для этой последней костяшки уберём вторую горизонтальную спичку из верхнего края данного квадрата. Затем рекурсивно решим головоломку для квадрата размера $n - 2$, который находится внутри меньшей из границ рамки.

Результат применения этого алгоритма к доскам $n \times n$ для чётных и нечётных значений n показан на рис. 4.104.

В двух словах, алгоритм работает потому, что средние линии домино, участвующих в покрытии соседних рамок, разбивают прямые линии внутри доски, и периметр оказывается разбитым в результате удаления одной спички. Это наблюдение можно легко формализовать с помощью математической индукции.

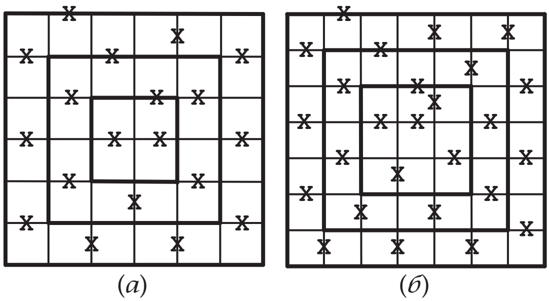


Рис. 4.104. «*Разрушение квадратов*» для (*а*) $n = 6$ и (*б*) $n = 7$

Покрытие костяшками домино также немедленно даёт нам формулу для числа $K(n)$ убираемых алгоритмом спичек. Действительно, если n чётно, то в покрытии участвуют $\frac{n^2}{2}$ костяшек домино; каждому домино соответствует одна удалённая спичка, кроме одной из тех домино, которые покрывают центральный квадрат 2×2, — ей соответствуют две. Таким образом, общее число убираемых спичек для чётных n равняется $\frac{n^2}{2} + 1$. Если $n > 1$ нечётно, то в покрытии рамок участвуют $\frac{n^2 - 1}{2}$ костяшек домино; каждому домино соответствует одна удалённая спичка, кроме одной из горизонтальных домино, которые покрывают центральный квадрат 3×3, — ей соответствуют две. Одна спичка дополнительно убирается из центрального квадрата 1×1. Таким образом, общее число убираемых спичек для нечётных n равняется $\frac{n^2 - 1}{2} + 2 = \left\lceil \frac{n^2}{2} \right\rceil + 1$.

При чётных n легко показать, что $K(n) = \left\lceil \frac{n^2}{2} \right\rceil + 1$ есть минимальное число спичек, которые нужно убрать, чтобы разрушить все квадраты. Если мы раскрасим доску в два цвета в шахматном порядке, то на ней будет $\frac{n^2}{2}$ тёмных клеток и $\frac{n^2}{2}$ светлых. Чтобы разрушить одиночные квадраты, соответствующие тёмным клеткам, нужно убрать хотя бы $\frac{n^2}{2}$ спичек. Этим же можно разрушить и все одиночные квадраты, соответствующие светлым клеткам, но только в том случае, если каждая удаляемая спичка лежит между тёмным и светлым квадратом. Так как таких спичек на периметре доски нет, нужно убрать ещё хотя бы одну спичку, так что итоговый минимум есть $\frac{n^2}{2} + 1$.

Если $n > 1$ нечётно, то несколько более хитрая цепочка рассуждений (см. [Gar06, p. 31–32]) показывает, что та же формула с дополнительным округлением вверх до ближайшего целого и в этом случае задаёт минимальное число спичек, которые нужно убрать.

Комментарии. Алгоритм решения головоломки, очевидно, основан на стратегии «уменьшай и властвуй». Интересно, что, решив задачу для $n = 1, 2$ и 3, легко прийти к неверному предположению, что ответом к задаче являются треугольные числа $(1, 3, 6, 10, \ldots)$.

Эту головоломку нашёл Мартин Гарднер в сборнике головоломок Сэма Лойда. Гарднер использовал её в своей колонке в *Scientific American* за ноябрь 1965 года, которую затем включил в книгу [Gar06, Problem 1.20].

145. Пятнашки

Решение. Если прочесть номера квадратиков, двигаясь сверху вниз и слева направо, можно связать список чисел от 1 до 15 с любой возможной ситуацией в игре. Цель при таком подходе — получить из начального списка

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 15,\ 14 \qquad (1)$$

его перестановку

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14,\ 15, \qquad (2)$$

совершив последовательно некоторое число разрешённых ходов.

Полезно рассмотреть *чётность* перестановок, определяющих позиции на доске. В общем случае, чтобы найти чётность перестановки, нужно найти число *инверсий* — так называются пары её элементов, стоящих в неправильном порядке: больший элемент стоит раньше меньшего.

Например, в перестановке 32154 инверсий четыре: $(3, 2)$, $(3, 1)$, $(2, 1)$ и $(5, 4)$. Поскольку 4 — чётное число, перестановку 32154 называют *чётной*. Перестановка же 23154 *нечётная*, поскольку в ней нечётное число инверсий: $(2, 1)$, $(3, 1)$ и $(5, 4)$. Приведём следующее очевидное, но немаловажное общее свойство чётности перестановок: если поменять местами два соседних элемента в перестановке, её чётность изменится на противоположную.

Возвращаясь к нашей головоломке, заметим, что у начальной и конечной позиций разная чётность: перестановка (1) нечётна, а (2) чётна. Теперь посмотрим, как чётность перестановки, соответствующей некоторому положению на доске, может измениться в результате игрового хода. В задаче разрешены два вида ходов: квадратик можно перемещать по горизонтали или вертикали на соседнюю свободную позицию. Перемещение по горизонтали не меняет перестановку и, следовательно, её чётность. Перемещение по вертикали порождает циклический сдвиг четырёх последовательных

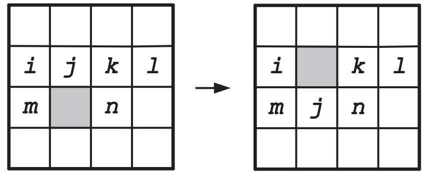


Рис. 4.105. Результат перемещения квадратика j вниз

элементов перестановки: например, квадратики j, k, l, m на рис. 4.105 после такого хода оказываются в порядке k, l, m, j.

Тот же циклический сдвиг получился бы, если бы мы трижды переставили соседние элементы этой четвёрки:

$$\ldots jklm \ldots \quad \to \quad \ldots kjlm \ldots \quad \to \quad \ldots kljm \ldots \quad \to \quad \ldots klmj \ldots$$

(Несмотря на то что так переставлять по правилам задачи нельзя, полезно использовать эти соображения, чтобы понять, как влияет перемещение по вертикали на чётность позиции.) Поскольку одна перестановка соседних номеров меняет чётность на противоположную, то же верно и для трёх перестановок. Удобно представлять ходы в игре как последовательность перемещений пустого квадрата. В нашей головоломке пустой квадрат находится на одном и том же месте и в начальном, и в конечном положениях. Так что и число горизонтальных, и число вертикальных перемещений в любой последовательности ходов, приводящей к решению задачи, должно быть чётным, чтобы на каждый сдвиг вправо приходился сдвиг влево, а на каждый сдвиг вверх — сдвиг вниз. Поскольку ни горизонтальные перемещения, ни чётное число вертикальных перемещений не меняют чётность позиции, чётности начальной и конечной позиций должны быть одинаковыми, чтобы у головоломки было решение. И поскольку это необходимое условие в нашем случае не выполнено, у головоломки решения нет.

Комментарии. Эта задача решается с помощью сравнения чётностей начальной и конечной позиций — стандартное применение идеи инварианта (во второй части учебного раздела можно найти её обсуждение и примеры применения). Заметим, что если добавить число строк к количеству инверсий, то чётность суммы останется неизменной и после вертикального перемещения квадратика. Один альтернативный вариант работы с пустым квадратиком — дать ему номер 16.

Ни для какой нечётной перестановки решения нет — в то же время для всякой чётной оно существует, хотя придумать эффективный алгоритм для его получения довольно трудно. В частности, доказано, что сложность нахождения кратчайшей последовательности ходов, приводящей к решению, крайне высока (задача NP-полна) для досок $n \times n$.

Пятнашки, также известные как «*Головоломка 14-15*» и «*Boss Puzzle*», — одна из наиболее известных головоломок, и редкий сборник головоломок обходится без неё. Эта загадка свела с ума весь мир в 1880 году; отчасти это подогревалось обещанием выплатить \$1000 любому, кто решит эту (нерешаемую) задачу. Изобретение головоломки часто приписывают Сэму Лойду (1841–1911), выдающемуся американскому изобретателю головоломок и знаменитому шахматному композитору. Однако это неверно, хотя и утверждалось самим Лойдом: головоломку придумал Ной Палмер Чепмэн, почтмейстер из Канастоты, штат Нью-Йорк, который подал заявку на патент в марте 1880 года. Заявка была отклонена, скорее всего, потому, что изобретение было похоже на уже запатентованное два года назад Эрнестом У. Кинси. Детали этой истории, а также множество других фактов, касающихся головоломки, можно найти в монографии [Slo06].

146. Стрельба по движущейся мишени

Решение. Для начала пронумеруем места, где может находиться мишень, слева направо от 1 до n.

Стрелку разумно сделать первый выстрел по позиции 2 (или по симметричной позиции $n-1$), поскольку только эти выстрелы могут гарантированно либо попасть по мишени, либо позволить быть уверенным в существовании места (1 или, соответственно, n), куда мишень *не может* пойти после первого выстрела. Сначала рассмотрим случай, когда мишень изначально была в позиции с чётным номером. Тогда после первого выстрела стрелок либо поразил цель, либо цель переместилась в позицию с нечётным номером, большим или равным 3. Поэтому, если стрелок направит свой второй выстрел на позицию 3, он либо попадёт по цели, либо будет уверен, что цель будет в позиции с чётным номером, большим или равным 4. Таким образом, если стрелок продолжит стрельбу последовательно по позициям $4, 5, \ldots, n-1$, то наверняка поразит мишень.

Если же перед первым выстрелом мишень находилась в позиции с нечётным номером, стрелок не попадёт по ней за $n-2$

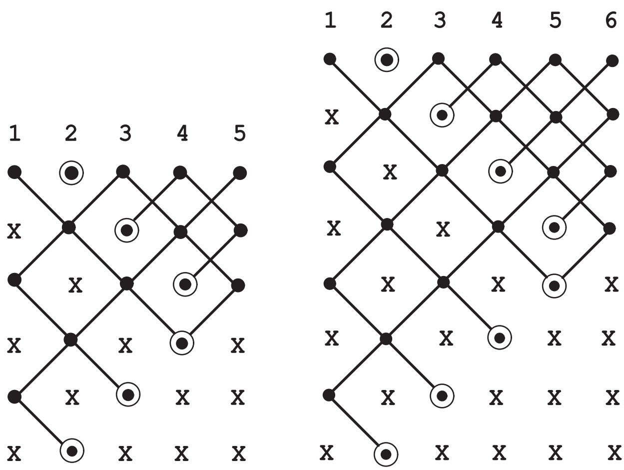


Рис. 4.106. Иллюстрация к алгоритму «*Стрельба по движущейся мишени*» для $n = 5$ и $n = 6$. Строка с номером i, $i = 1,\ldots,2n-4$, изображает возможные (маленькие чёрные кружки) и не возможные (X) места нахождения движущейся мишени перед i-м выстрелом; сам выстрел производится по обведённой позиции

вышеописанных выстрелов, поскольку позиции цели и выстрелов всегда будут иметь разные чётности. Но после выстрела по позиции $n-1$ цель переместится в позицию с той же чётностью, что и у $n-1$. Поэтому, если симметрично продолжить последовательность выстрелов $n-1, n-2,\ldots,2$, то стрелку гарантируется попадание по мишени и в этом случае.

В итоге последовательность из $2(n-2)$ выстрелов по позициям с номерами

$$2,\ 3,\ldots,\ n-1,\ n-1,\ n-2,\ \ldots,2$$

гарантирует попадание по мишени для любого $n > 2$. Если же $n = 2$, задачу решат два выстрела по одной и той же позиции.

Решение можно проиллюстрировать, а на самом деле и вывести из диаграмм, изображённых на рис. 4.106. Эти диаграммы показывают, где мишень может быть и где её быть не может перед каждым выстрелом, для $n = 5$ и $n = 6$.

Комментарии. Число позиций, в которых всё ещё может быть мишень, выступает в этом решении в роли полуинварианта. С другой стороны, на использование диаграмм рис. 4.106, конечно, можно смотреть как на отличный пример стратегии замены представления.

Частный случай задачи, с $n = 1000$, предлагался на российской математической олимпиаде в 1999 году и был опубликован в журнале «Квант» в 2000 г. (№ 2, с. 21).

147. Шапки с номерами

Решение. Математики могут выиграть следующим образом.

В самом начале они присваивают себе номера от 0 до $n-1$, например сообразно с алфавитным порядком их имён. Затем, видя числа на шапках всех остальных математиков, i-й математик, $0 \leqslant i \leqslant n-1$, считает сумму этих чисел S_i и предполагает, что на его/её шапке написано наименьшее неотрицательное решение уравнения

$$(S_i + x_i) \bmod n = i,$$

которое есть

$$x_i = (i - S_i) \bmod n.$$

(Другими словами, x_i вычисляется как остаток от деления на общее число математиков разности между номером самого математика и суммой чисел на шапках всех остальных.)

Пусть $S = h_1 + h_2 + \ldots + h_n$ есть сумма чисел на всех шапках. Очевидно, $S = S_i + h_i$ для любого $0 \leqslant i \leqslant n-1$. Поскольку остаток от деления произвольного числа на n пробегает все неотрицательные целые числа от 0 до $n-1$ включительно, существует ровно одно целое j такое, что

$$j \bmod n = S \bmod n.$$

И j-й математик в таком случае угадывает своё настоящее число. Действительно,

$$j = j \bmod n = S \bmod n = (S_j + h_j) \bmod n = (S_j + x_j) \bmod n.$$

Отсюда $h_j \bmod n = x_j \bmod n$, и поскольку и h_j, и x_j лежат между 0 и $n-1$, $x_j = h_j$.

Например, пусть $n = 5$, а на шапках написаны числа $3, 4, 0, 3, 2$. Тогда имеем следующую таблицу:

i	h_i	S_i	x_i	Верно ли угадано число
0	3	9	1	нет
1	4	8	3	нет
2	0	12	0	да ($j = 2$)
3	3	9	4	нет
4	2	10	4	нет

Комментарии. В последние годы головоломка появилась под названиями «*Rainbow Hats*» и «*88 Hats*» на нескольких интернет-сайтах и в книгах, посвящённых вопросам, которые задают на собеседованиях при приёме на работу по техническим специальностям (см., например, [Zho08, p. 31]). Мы не смогли определить, где она появилась впервые.

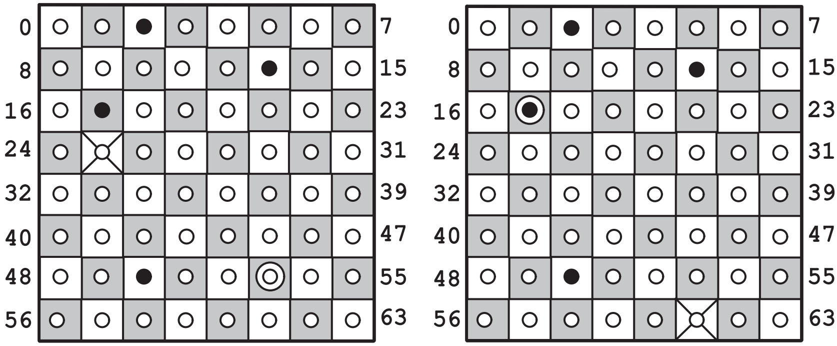
148. Свобода за одну монету

Решение. Заключённым необходимо придумать способ, как заключённому A, перевернув одну монету, сообщить тем самым заключённому B, какую клетку выбрал тюремщик. Чтобы это сделать, им нужно воспользоваться знанием о том, какие монеты лежат, скажем, решками вверх. А именно, им следует найти функцию, которая по позициям всех лежащих вверх решкой монет вычисляет позицию выбранной клетки. Задача A — перевернуть одну монету так, чтобы задать это соответствие; задача B — просто вычислить значение функции для доски, лежащей перед ним. Вот как это можно сделать.

Во-первых, пронумеруем клетки доски от 0 до 63, например двигаясь слева направо и сверху вниз. Пусть $T_1, T_2, \ldots, T_n$ — двоичное 6-битовое представление последовательности номеров всех клеток доски, на которых монеты лежат решкой вверх и которую изначально видит A; пусть J — 6-битовое двоичное представление номера клетки, которую выбрал тюремщик. Пусть X — 6-битовое двоичное представление номера клетки, которую должен перевернуть A. Чтобы найти X, вычислим исключающее «ИЛИ» (XOR, обозначается знаком $\oplus$) суммы $T = T_1 \oplus T_2 \oplus \ldots \oplus T_n$ и J:

$$T \oplus X = J, \quad \text{или} \quad X = T \oplus J. \tag{1}$$

(Если $n = 0$, мы считаем, что $T = O$, т. е. строка из всех нулей, и, следовательно, $X = O \oplus J = J$.)



(*а*) (*б*)

Рис. 4.107. Два частных случая задачи «*Свобода за одну монету*». Монеты, изначально лежащие решкой вверх и орлом вверх, обозначены соответственно чёрными и белыми кружками; выбранная для угадывания монета обозначена крестом. Вокруг монеты, которую должен перевернуть первый заключённый, нарисован дополнительный круг

Например, для доски на рис. 4.107, *а* имеем:

$$\begin{array}{r} T_1 = 2_{10} = 000010 \\ T_2 = 13_{10} = 001101 \\ T_3 = 17_{10} = 010001 \\ T_4 = 50_{10} = 110010 \\ \hline T = 101100 \end{array} \qquad J = 25_{10} = 011001$$

И, следовательно,

$$X = T \oplus J = 101100 \oplus 011001 = 110101 = 53_{10}.$$

Так что, после того как заключённый A перевернёт монету на клетке 53 решкой вверх, заключённый B увидит доску, на которой монеты на клетках 2, 13, 17, 50 и 53 лежат решками вверх, и сможет вычислить положение выбранной клетки как

$$T_1 \oplus T_2 \oplus \cdots \oplus T_n \oplus X = T \oplus X = J = 011001 = 25_{10}.$$

Вышеприведённый пример касается одного из двух возможных случаев — того, при котором монета на клетке X, вычисляемой по формуле (1), лежит орлом вверх. Тогда, если её перевернуть, X действительно добавится к остальным лежащим решкой вверх монетам. Но что если монета в этой клетке уже лежит решкой вверх, т. е. является i-й лежащей решкой вверх монетой для некоторого $1 \leqslant i \leqslant n$?

В этом случае, если эту монету перевернуть, она будет лежать орлом вверх, и заключённому В останется вычислить позицию выбранной клетки как $T_1 \oplus \ldots \oplus T_{i-1} \oplus T_{i+1} \oplus \ldots \oplus T_n$. К счастью, формула (1) работает и в этом случае, поскольку $S \oplus S = O$ для любой битовой строки S. Действительно, если заключённый А вычисляет $X = T \oplus J = T_i$, то заключённый В получит ту же позицию выбранной клетки, которую использовал в своих вычислениях заключённый А:

$$\begin{aligned} J = T \oplus X = T_1 \oplus \cdots \oplus T_{i-1} \oplus T_i \oplus T_{i+1} \oplus \cdots \oplus T_n \oplus T_i = \\ = T_1 \oplus \cdots \oplus T_{i-1} \oplus T_{i+1} \oplus \cdots \oplus T_n \oplus T_i \oplus T_i = \\ = T_1 \oplus \cdots \oplus T_{i-1} \oplus T_{i+1} \oplus \cdots \oplus T_n. \end{aligned}$$

Например, если тюремщик выбирает клетку 61 на доске с теми же четырьмя лежащими решкой вверх монетами (см. рис. 4.107, *б*), то

$$\begin{array}{r} T_1 = \ \ 2_{10} = 000010 \\ T_2 = 13_{10} = 001101 \\ T_3 = 17_{10} = 010001 \\ T_4 = 50_{10} = 110010 \\ \hline T = 101100 \end{array} \qquad J = 61_{10} = 111101$$

и

$$X = T \oplus J = 101100 \oplus 111101 = 010001 = T_3 = 17_{10}.$$

После того как заключённый А перевернёт монету на клетке 17 орлом вверх, заключённый В увидит доску, на которой решкой вверх лежат монеты на клетках 2, 13 и 50, и вычислит положение выбранной клетки как

$$000010 \oplus 001101 \oplus 110010 = 111101 = 61_{10}.$$

Комментарии. Решение головоломки даёт ещё один пример того, как порой полезно использовать двоичную систему счисления.

Одномерный вариант этой задачи предлагался на Международном математическом турнире городов осенью 2007 года, но там от участников не требовалось представить алгоритм решения целиком. С тех пор головоломка появлялась на нескольких интернет-сайтах в той форме, которая приведена выше.

149. Распространение камушков

Решение. Головоломка имеет решение только при $n = 1$ и $n = 2$.

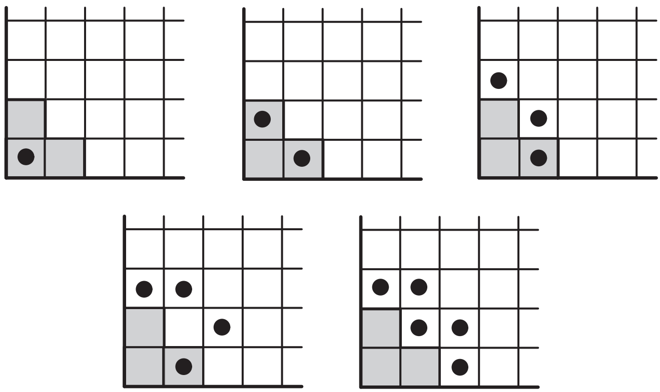


Рис. 4.108. Последовательность ходов, в результате которой от камушков освобождается «лестница» S_2 (изображена тёмным) в головоломке «*Распространение камушков*»

Единственный допустимый первый ход приводит к решению задачи при $n=1$. На рис. 4.108 показано решение при $n=2$.

Теперь покажем, что ни для какого $n>2$ «лестницу» S_n нельзя очистить от камушков с помощью конечного числа допустимых ходов. С этой целью присвоим каждой клетке (i,j) на доске, $i,j\geqslant 1$, вес $w(i,j)=2^{2-i-j}$ (рис. 4.109).

Заметим, что все клетки на диагонали с номером d, который определяется соотношением $i+j=d+1$, $d=1,2,\ldots$, имеют одинаковый вес, а сумма весов всех клеток — которую можно вычислить, просуммировав веса в каждом ряду — равняется 4:

$$\left(1+\frac{1}{2}+\frac{1}{4}+\cdots\right)+\left(\frac{1}{2}+\frac{1}{4}+\cdots\right)+\cdots+\left(\frac{1}{2^{j-1}}+\frac{1}{2^j}+\cdots\right)+\cdots=$$

$$=\frac{1}{1-\frac{1}{2}}+\frac{\frac{1}{2}}{1-\frac{1}{2}}+\cdots+\frac{\frac{1}{2^{j-1}}}{1-\frac{1}{2}}+\cdots=2\left(1+\frac{1}{2}+\cdots\frac{1}{2^{j-1}}+\cdots\right)=$$

$$=2\cdot\frac{1}{1-\frac{1}{2}}=4.$$

Затем можно определить вес данной позиции на доске как сумму весов всех занятых камушками клеток. Вес начальной позиции равен 1 для любого $n\geqslant 1$. Ни один ход — а следовательно, и никакая конечная последовательность ходов —

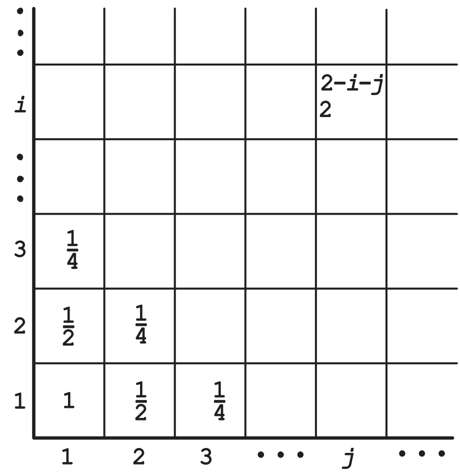


Рис. 4.109. Веса, которые присваиваются клеткам доски в задаче «*Распространение камушков*»

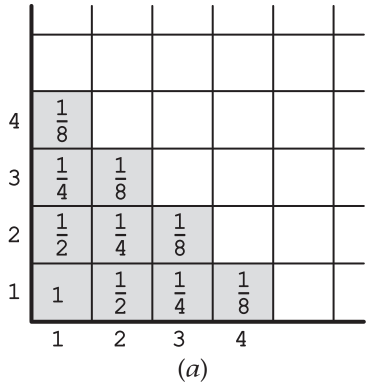
не меняет вес позиции, поскольку вес камушка, который мы убираем с клетки (i,j), компенсируется весами новых камушков на клетках $(i+1,j)$ и $(i,j+1)$: $2^{2-(i+j)} = 2^{2-(i+1+j)} + 2^{2-(i+j+1)}$.

Отсюда немедленно следует, что невозможно освободить от камушков никакую «лестницу» S_n для $n \geqslant 4$ в результате конечного числа ходов. Если бы это было возможно, то в конечной позиции камушки занимали бы некоторую область R_n, состоящую из конечного числа клеток вне S_n. Общий вес этой области $W(R_n)$ должен был бы быть меньше, чем общий вес всех клеток вне S_4. Последний можно найти, вычтя вес клеток, составляющих S_4, из веса всех клеток доски (рис. 4.110, *а*):

$$W(R_4) < 4 - W(S_4\ ) = 4 - \left(1 + 2\cdot\frac{1}{2} + 3\cdot\frac{1}{4} + 4\cdot\frac{1}{8}\right) = \frac{3}{4}.$$

Таким образом, поскольку $W(R_n) \leqslant W(R_4) < 1$ для $n \geqslant 4$, никакую «лестницу» S_n при $n \geqslant 4$ нельзя освободить от камушков в результате конечного числа ходов.

«Лестницу» S_3 освободить также нельзя. Чтобы это доказать, заметим, что первая строка и первый столбец доски всегда содержат по одному камушку. Поэтому, если бы все камушки из S_3 перешли в некоторую область R_3 за пределами S_3, R_3 состояла бы из одной клетки в первой строке, одной клетки в первом столбце и набора клеток Q_3 в других строках и столбцах (рис. 4.110, *б*).



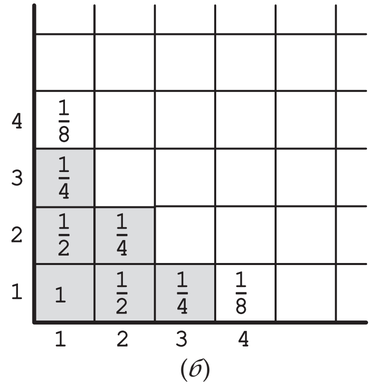


Рис. 4.110. (*а*) Веса клеток «лестницы» S_4 и (*б*) веса клеток «лестницы» S_3 и ещё двух внешних клеток строки 1 и столбца 1 в головоломке «*Распространение камушков*»

Верхнюю границу для значения $W(Q_3)$ можно вычислить как разность между весом всего квадранта и весами клеток в первом столбце, клеток в первой строке, а также клетки $(2,2)$:

$$W(Q_3) < 4 - \left[1 + 2\left(\frac{1}{2} + \frac{1}{4} + \ldots\right) + \frac{1}{4}\right] = \frac{3}{4}.$$

Отсюда получаем следующее ограничение сверху на $W(R_3)$:

$$W(R_3) < \frac{1}{8} + \frac{1}{8} + \frac{3}{4} = 1.$$

Поскольку $W(R_3) < 1$, камушки из S_3 не могут распространиться так, чтобы занять только R_3.

Комментарии. Идея инварианта — инвариантен вес позиции на доске — в этом решении схожа с той, которую Дж. Конвей применил в головоломке «*Солдаты Конвея*» (№ 132).

Эту головоломку предложил М. Концевич в ноябрьском номере журнала «Квант» 1981 года (с. 21, задача М715); А. Ходулёв представил подробное решение задачи, а также обобщил её в июльском выпуске того же журнала 1982 года.

150. Болгарский пасьянс

Решение. Каждую кучку разумно представлять как столбик монет. На каждой итерации алгоритма мы идём по ряду таких столбиков, берём одну монету из каждого — без ограничения общности можно считать, что монета берётся снизу столбика — и помещаем эти монеты в новый столбик в том порядке, в котором брали. Сначала мы поместим новый столбик перед другими, а затем, если он меньше следующего, поставим его на подходящее место в ряду кучек, чтобы в итоге ряд был отсортирован в порядке невозрастания размеров кучек. Этот процесс показан на рис. 4.111 — монеты обозначены буквами.

Поскольку число способов представить n как сумму положительных целых чисел (размеров кучек) конечно, алгоритм обязательно зациклится после конечного числа итераций. Наша задача — показать, что вне зависимости от разбиения числа $n = 1 + 2 + \ldots + k$, включая и то, при котором в начале есть одна кучка из n монет, алгоритм в любом случае рано или поздно приведёт к разбиению $(k, k-1, \ldots, 1)$. Такой способ разделения монет есть цикл из одного состояния: алгоритм преобразует его в самого себя. Других циклов из одного состояния нет. Действительно, если шаг алгоритма преобразует $(n_1, n_2, \ldots, n_s)$, где $n_1 \geqslant n_2 \geqslant \ldots \geqslant n_s$, в $(s, n_1 - 1, n_2 - 1, \ldots, n_{s-1} - 1)$ и оба набора совпадают, то s должно быть максимальным числом во втором наборе, а n_s должно равняться 1 (в противном случае наборы были бы разного размера). Приравнивая соответствующие компоненты наборов, получаем систему линейных уравнений $n_1 = s$ и $n_i = n_{i-1} - 1$ для $i = 2, \ldots, s$, которую легко решить с помощью обратной подстановки: $n_{s-1} = n_s + 1 = 2$, и т. д. до $n_1 = n_2 + 1 = k$ и $s = k$.

Теперь покажем, что не существует цикла, который состоял бы из более чем одного разбиения.

Во-первых, внутри цикла не может происходить пересортировка кучек. Это утверждение можно доказать, присвоив каждой монете вес, равный сумме номера её столбца и её позиции в столбце, считая снизу. Например, в начальной позиции на рис. 4.111 монетам присваиваются следующие веса: $a(1+1), b(1+2), \ldots, j(4+1)$. Затем можно определить вес разбиения (не обязательно отсортированного) как сумму весов всех составляющих его монет. Например, вес начального разбиения на рис. 4.111 равен 41. Легко видеть, что шаг алгоритма не меняет вес разбиения, если не пересортировывать результат: вес $i + j$ монеты в столбике i на позиции j становится равным $(i+1) + (j-1)$ при $j > 1$ или $j + i$ при $j = 1$ соответственно. Но если столбики оказываются не расположенными в порядке невозрастания их размеров, то их

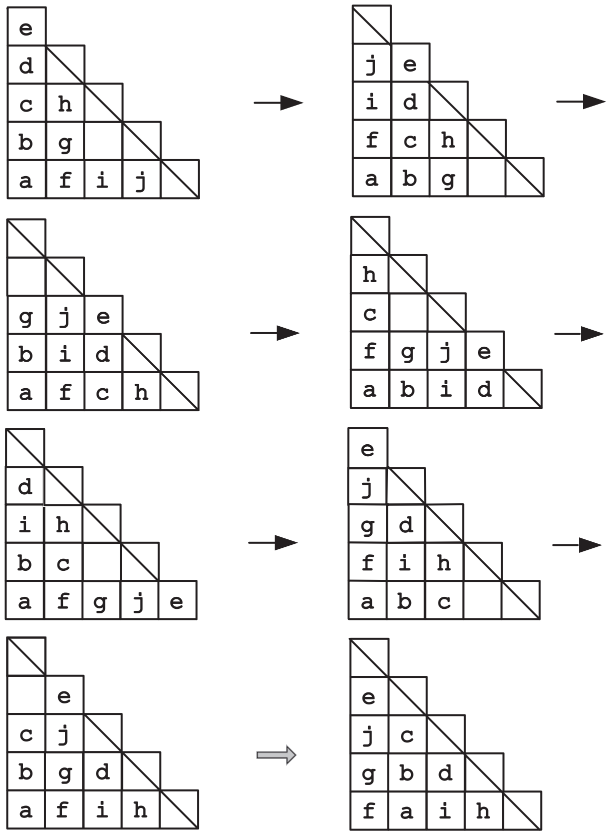


Рис. 4.111. Пример головоломки «*Болгарский пасьянс*». На последнем шаге приходится отсортировать сами кучки

сортировка уменьшает вес разбиения. Следовательно, внутри цикла не может происходить такая сортировка.

Во-вторых, в любой последовательности разбиений, которые получаются в результате применения алгоритма и не требуют сортировки, каждая монета циркулирует вдоль своей диагонали, так же, как и пустые позиции на диагонали, если такие есть (пример см. на рис. 4.111). Но тогда, если на диагонали с номером d есть хотя бы одна монета, на диагонали с номером $d-1$ не может быть пустых позиций. Действительно, если бы это было возможно, пустая позиция на диагонали с номером $d-1$ и монета на диагонали с номером d после конечного числа шагов оказались бы на одной высоте в своих столбцах — а в этой ситуации понадобилось бы отсортировать столбцы (см. последний шаг на рис. 4.111).

Отсюда немедленно следует, что лишь на самой большой диагонали может недоставать монет, если речь идёт о последовательности состояний, образующих цикл. Но если $n = 1+2+\ldots+k$, то все k диагоналей должны быть заполнены монетами. Действительно, если бы на k-й диагонали не было монет, то общее число монет не могло бы быть больше, чем $1+2+\ldots+(k-1)$. А если бы на $(k+1)$-й диагонали была монета, то, как доказано выше, все диагонали меньшего размера были бы заполнены и общее число монет превысило бы $1+2+\ldots+k$. Таким образом, для таких n цикл может состоять только из одного разбиения.

Комментарии. Популярность болгарскому пасьянсу принёс Мартин Гарднер, опубликовав его в своей колонке в *Scientific American* в 1983 году (см. также [Gar97b, p. 36–43]). Она основывалась на статье датского математика Йоргена Брандта, опубликованной годом ранее. С тех пор было получено несколько интересных результатов об этой игре и её вариациях (см., например, [Gri98]). В частности, было доказано, что для достижения стабильного разбиения $(k, k-1, \ldots, 1)$ из любого начального состояния требуется не более k^2-k итераций.

СПИСОК ЛИТЕРАТУРЫ

[Ash04] Ash, J. M., and Golomb, S. W. Tiling deficient rectangles with trominoes. *Mathematics Magazine*, vol. 77, no. 1 (Feb. 2004), 46–55.

[Ash90] Asher, M. A river-crossing problem in cross-cultural perspective. *Mathematics Magazine*, vol. 63, no. 1 (Feb. 1990), 26–29.

[Ave00] Averbach, B., and Chein, O. *Problem Solving Through Recreational Mahtematics*. Dover, 2000.

[Bac12] Bachet, C. *Problèmes plaisans et delectables qui se font par les nombres*. Paris, 1612.

[Backh] Backhouse, R. *Algorithmic problem solving course website*. www.cs.nott.ac.uk/~rcb/G51APS/exercises/InductionExercises.pdf (accessed Oct. 4, 2010).

[Bac08] Backhouse, R. The capacity-C torch problem. *Mathematics of Program Construction 9th International Conference* (*MPC* 2008), Marseille, France, July 15–18, 2008, Springer-Verlag, 57–78.

[Bal87] Ball, W. W. Rouse, and Coxeter, H. S. M. *Mathematical Recreations and Essays*, 13th edition. Dover, 1987. www.gutenberg.org/ebooks/26839 (1905 edition; accessed Oct. 10, 2010).[1]

[Bea92] Beasley, J. D. *The Ins and Outs of Peg Solitaire*. Oxford University Press, 1992.

[Bec97] Beckwith, D. Problem 10459, in Problems and Solutions, *American Mathematical Monthly*, vol. 104, no. 9 (Nov. 1997), 876.

[Bel09] Bell, J., and Stevens, B. A survey of known results and research areas for n-queens. *Discrete Mathematics*, vol. 309, issue 1 (Jan. 2009), 1–31.

[Ben00] Bentley, J. *Programming Pearls*, 2nd ed. Addison-Wesley, 2000.[2]

[1] Русский перевод: *Болл У., Коксетер Г.* Математические эссе и развлечения. — М.: Мир, 1986.

[2] Русский перевод: *Бентли Дж.* Жемчужины программирования. — Спб.: Питер, 2002.

[Ber04] Berlekamp, E. R., Conway, J. H., and Guy, R. K. *Winning Ways for Your Mathematical Plays*, Volume 4, 2nd ed. A K Peters, 2004.

[Ber91] Bernhardsson, B. Explicit solutions to the *n*-queens problem for all *n*. *SIGART Bulletin*, vol. 2, issue 2 (April 1991), 7.

[Bogom] Bogomolny, A. *Interactive Mathematics Miscellany and Puzzles*. www.cut-the-knot.org (accessed Oct. 4, 2010).

[Bog00] Bogomolny, A. The three jugs problem. The Mathematical Association of America, May 2000. www.maa.org/editorial/knot/water.html#kasner (accessed Oct. 10, 2010).

[Bol07] Bollobas, B. *The Art of Mathematics: Coffee Time in Memphis*. Cambridge University Press, 2007.

[Bos07] Bosova, L. L., Bosova, A. Yu., and Kolomenskaya, Yu. G. *Entertaining Informatics Problems*, 3rd ed., BINOM, 2007 (in Russian).[1)]

[Bro63] Brooke, M. *Fun for the Money*. Charles Scribner's Sons, 1963.

[CarTalk] Archive of the U. S. National Public Radio talk show *Car Talk*. www.cartalk.com/content/puzzler (accessed Oct. 4, 2010).

[Chr84] Christen, C., and Hwang, F. Detection of a defective coin with a partial weight information. *American Mathematical Monthly*, vol. 91, no. 3 (March 1984), 173–179.

[Chu87] Chu, I-Ping, and Johnsonbaugh, R. Tiling and recursion. *ACM SIGCSE Bulletin*, vol. 19, issue 1 (Feb. 1987), 261–263.

[Cor09] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms*, 3rd edition. MIT Press, 2009.[2)]

[Cra07] Crack, T. F. *Heard on the Street: Quantitative Questions from Wall Street Job Interviews*, 10th ed. Self-published, 2007.

[Cso08] Csorba, P., Hurkens, C. A., and Woeginger, G. J. The Alcuin number of a graph. *Proceedings of the 16th Annual European Symposium on Algorithms. Lecture Notes in Computer Science*, vol. 5193, 2008, 320–331.

[Dem02] Demaine, E. D., Demaine, M. L., and Verrill, H. Coin-moving puzzles. In R. J. Nowakowski, editor, *More Games of No Chance*. Cambridge University Press, 2002, 405–431.

1) *Босова Л. Л., Босова А. Ю., Коломенская Ю. Г.* Занимательные задачи по информатике. — 3-е изд. — М.: БИНОМ. Лаборатория знаний, 2007.

2) Русский перевод: *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы. Построение и анализ. — М.: Вильямс, 2015.

[Dij76] Dijkstra, E. W. *A Discipline of Programming*. Prentice Hall, 1976.[1)]

[Dud02] Dudeney, H. E. *The Canterbury Puzzles and Other Curious Problems*. Dover, 2002. www.gutenberg.org/ebooks/27635 (1919 edition; accessed Oct. 10, 2010).[2)]

[Dud58] Dudeney, H. E. *Amuzements in Mathematics*. Dover, 1958. www.gutenberg.org/ebooks/16713 (first published in 1917; accessed Oct. 10, 2010).

[Dud67] Dudeney, H. E. (edited by Martin Gardner). *536 Puzzles & Curious Problems*. Charles Scribner's Sons, 1967.[3)]

[Dyn71] Dynkin, E. B., Molchanov, S. A., Rozental, A. L., and Tolpygo, A. K. *Mathematical Problems*, 3rd revised edition, Nauka, 1971 (in Russian).[4)]

[Eng99] Engel, A. *Problem-Solving Strategies*. Springer, 1999.

[Epe70] Eperson, D. B. Triangular (Old) Pennies. *The Mathematical Gazette*, vol. 54, no. 387 (Feb. 1970), 48–49.

[Fom96] Fomin, D., Genkin, S., and Itenberg, I. *Mathematical Circles (Russian Experience)*. American Mathematical Society, Mathematical World, Vol. 7, 1996 (translated from Russian).[5)]

[Gar99] Gardiner, A. *Mathematical Puzzling*. Dover, 1999.

[Gar61] Gardner, M. *Mathematical Puzzles*. Thomas Y. Crowell, 1961.

[Gar71] Gardner, M. *Martin Gardner's 6th Book of Mathematical Diversions from Scientific American*. W. H. Freeman, 1971.[6)]

[Gar78] Gardner, M. *aha! Insight*. Scientific American / W. H. Freeman, 1978.[7)]

1) Русский перевод: *Дейкстра Э.* Дисциплина программирования. — М.: Мир, 1978.

2) Русский перевод: *Дьюдени Г. Э.* Кентерберийские головоломки. — М.: Мир, 1979. (1-е изд.) — М.: Зебра-Е, 2011. (2-е изд.)

3) Русский перевод: *Дьюдени Г. Э.* 520 головоломок. — М.: Мир, 1975.

4) *Дынкин Е. Б., Молчанов С. А., Розенталь А. Л, Толпыго А. К.* Математические задачи. — 3-е изд. — М.: Наука, 1971.

5) *Генкин С. А., Итенберг И. В., Фомин Д. В.* Ленинградские математические кружки: пособие для внеклассной работы. — Киров: Издательство «АСА», 1994.

6) Русский перевод: *Гарднер М.* Математические досуги. — М.: Мир, 1972; 2000. — Гл. 26–37; *Гарднер М.* Математические новеллы. — М.: Мир, 1974; 2000. — Гл. 1–6.

7) Русский перевод: *Гарднер М.* Есть идея! — М.: Мир, 1982.

[Gar83] Gardner, M. *Wheels, Life, and Other Mathematical Amusements.* W. H. Freeman, 1983.[1)]

[Gar86] Gardner, M. *Knotted Doughnuts and Other Mathematical Entertainments*. W. H. Freeman, 1986.

[Gar87] Gardner, M. *The Second Scientific American Book of Puzzles and Games*. University of Chicago Press, 1987.[2)]

[Gar88a] Gardner, M. *Hexaflexagons and Other Mathematical Diversions: The First Scientific American Book of Puzzles and Games*. University of Chicago Press, 1988.[3)]'

[Gar88b] Gardner, M. *Time Travel and Other Mathematical Bewilderments*. W. H. Freeman, 1988.[4)]

[Gar89] Gardner, M. *Mathematical Carnival*. The Mathematical Association of America, 1989.[5)]

[Gar97a] Gardner, M. *Penrose Tiles to Trapdoor Chiphers ... and the Return of Dr. Matrix*, revised edition. The Mathematical Association of America, 1997.[6)]

[Gar97b] Gardner, M. *The Last Recreations: Hidras, Eggs, and Other Mathematical Mystifications*. Springer, 1997.

[Gar06] Gardner, M. *Colossal Book of Short Puzzles and Problems*. W. W. Norton, 2006.

[Gik76] Gik, E. Ya. *Mathematics on the Chessboard*. Nauka, 1976 (in Russian).[7)]

[Gik80] Gik, E. The Battleship game. *Kvant*, Nov. 1980, 30–32, 62–63 (in Russian).[8)]

[Gin03] Ginat, D. The greedy trap and learning from mistakes. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, ACM, 2003, 11–15.

[1)] Русский перевод: *Гарднер М.* Крестики-нолики. — М.: Мир, 1988.

[2)] Русский перевод: *Гарднер М.* Математические головоломки и развлечения. — М.: Мир, 1971. — Гл.16–34.

[3)] Русский перевод: *Гарднер М.* Математические головоломки и развлечения. — М.: Мир, 1971. — Гл. 1–5.

[4)] Русский перевод: *Гарднер М.* Путешествие во времени. — М.: Мир, 1990.

[5)] Русский перевод: *Гарднер М.* Математические новеллы. — М.: Мир, 1974 (отдельные главы); *Гарднер М.* Нескучная математика. — М.: АСТ, 2009.

[6)] Русский перевод: *Гарднер М.* От мозаик Пенроуза к надежным шифрам. — М.: Мир, 1993.

[7)] *Гик Е. Я.* Математика на шахматной доске. — М.: Наука, 1976.

[8)] *Гик Е.* Морской бой // Квант. — 1980. — № 11. — С. 30–32, 62–63.

[Gin06] Ginat, D. Coloful Challenges column. *inroads — SIGCSE Bulletin*, vol. 38, no. 2 (June 2006), 21–22.

[Gol54] Golomb, S. W. Checkerboards and polyominoes. *American Mathematical Monthly*, vol. 61, no. 10 (Dec. 1954), 675–682.

[Gol94] Golomb, S. W. *Polyominoes: Puzzles, Patterns, Problems, and Packings*. 2nd edition. Princeton University Press, 1994.[1]

[Graba] Grabarchuk, S. Coin triangle. From *Puzzles.com*. www.puzzles.com/PuzzlePlayground/CoinTriangle/CoinTriangle.htm (accessed Oct. 4, 2010).[2]

[Gra05] Grabarchuk, S. *The New Puzzle Classics: Ingenious Twists on Timeless Favorites*. Sterling Publishing, 2005.

[Gra94] Graham, R. L., Knuth, D. E. and Patashnik, O. *Concrete Mathematics: A Foundation for Computer Science*, 2nd ed. Addison-Wesley, 1994.[3]

[Gre73] Greenes, C. E. Function generating problems: the row chip switch. *Arithmetic Teacher*, vol. 20 (Nov. 1973), 545–549.

[Gri98] Griggs, J. R., and Ho, Chih-Chang. The cycling of partitions and compositions under repeated shifts. *Advances in Applied Mathematics*, vol. 21, no. 2 (1998), 205–227.

[Had92] Hadley, J., and Singmaster, D. Problems to sharpen the young. *Mathematical Gazette*, vol. 76, no. 475 (March 1992), 102–126.

[Hes09] Hess, D. *All-Star Mathlete Puzzles*. Sterling, 2009.

[Hof79] Hofstadter, D. *Gödel, Escher, Bach: An Eternal Golden Braid*. Basic Books, 1979.[4]

[Hur00] Hurkens, C. A. J. Spreading gossip efficiently. *NAW*, vol. 5/1 (June 2000), 208–210.

[Iba03] Iba, G., and Tanton, J. Candy sharing. *American Mathematical Monthly*, vol. 110, no. 1 (Jan. 2003), 25–35.

[Ign78] Ignat'ev, E. I. *In the Kindom of Quick Thinking*. Nauka, 1978 (in Russian).[5]

[1] Русский перевод первого издания книги: *Голомб С.* Полимино. — М.: Мир, 1975.

[2] На русском языке см. также http://puzzlepedia.ru/golsmonetami.html. — *Прим. ред.*

[3] Русский перевод: *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. Математические основы информатики. — М.: БИНОМ. Лаборатория знаний, 2009.

[4] Русский перевод: *Хофштадтер Д.* Гёдель, Эшер, Бах: эта бесконечная гирлянда. — Самара: Бахрах-М, 2001.

[5] *Игнатьев Е. И.* В царстве смекалки. — М.: Наука, 1978.

[Iye66] Iyer, M., and Menon, V. On coloring the $n \times n$ chessboard. *American Mathematical Monthly*, vol. 73, no. 7 (Aug.–Sept. 1966), 721–725.

[Kho82] Khodulev, A. Relocation of chips. *Kvant*, July 1982, 28–31, 55 (in Russian).[1)]

[Kin82] King, K. N., and Smith-Thomas, B. An optimal algorithm for sink-finding. *Information Processing Letters*, vol. 14, no. 3 (May 1982), 109–111.

[Kle05] Kleinberg, J., and Tardos, E. *Algorithm Design*. Addison-Wesley, 2005.[2)]

[Knott] Knott, R. *Fibonacci Numbers and the Golden Section*. www.mcs.surrey.ac.uk/Personal/R.Knott/Fibonacci/ (accessed Oct. 4, 2010).

[Knu97] Knuth, D. E. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, 3rd ed. Addison-Wesley, 1997.[3)]

[Knu98] Knuth, D. E. *The Art of Computer Programming, Volume 3: Sorting and Searching*, 2nd ed. Addison-Wesley, 1998.[4)]

[Knu11] Knuth, D. E. *The Art of Computer Programming, Volume 4A, Combinatorial Algorithms, Part 1*. Pearson, 2011.[5)]

[Kon96] Konhauser J. D. E., Velleman, D., and Wagon, S. *Which Way Did the Bicycle Go?: And Other Intriguing Mathematical Mysteries*. The Dolciani Mathematical Expositions, No. 18, The Mathematical Association of America, 1996.

[Kor72] Kordemsky, B. A. *The Moscow Puzzles: 359 Mathematical Recreations*. Scribner, 1972 (translated from Russian).[6)]

[1)] *Ходулёв А.* Расселение фишек // Квант. — 1982. — №7. — С. 28–31, 55.

[2)] Русский перевод: *Клейнберг Дж., Тардос Е.* Алгоритмы: разработка и применение. — Спб.: Питер, 2016.

[3)] Русский перевод: *Кнут Д. Э.* Искусство программирования. Том 1. Основные алгоритмы. — 3-е изд. — М.: Вильямс, 2015.

[4)] Русский перевод: *Кнут Д. Э.* Искусство программирования. Том 3. Сортировка и поиск. — 2-е изд. — М.: Вильямс, 2012.

[5)] Русский перевод: *Кнут Д. Э.* Искусство программирования. Том 4, А. Комбинаторные алгоритмы. Часть 1. — М.: Вильямс, 2015.

[6)] Русский оригинал: *Кордемский Б. А.* Математическая смекалка. — М.: Гостехиздат, 1954. (1-е изд.) — М.: Альпина Паблишер, 2016.

[Kor05] Kordemsky, B. A. *Mathematical Charmers*. Oniks, 2005 (in Russian).[1)]

[Kra53] Kraitchik, M. *Mathematical Recreations*, 2nd revised edition. Dover, 1953.

[Kre99] Kreher, D. L., and Stinson, D. R. *Combinatorial Algorithms: Generation, Enumeration, and Search*. CRC Press, 1999.

[Kur89] Kurlandchik, L. D., and Fomin, D. B. Etudes on the semi-invariant. *Kvant*, no. 7, 1989, 63–68 (in Russian).[2)]

[Laa10] Laakmann, G. *Cracking the Coding Interview*, 4th ed. CareerCup, 2010.

[Leh65] Lehmer, D. H. Permutation by adjacent interchanges. *American Mathematical Monthly*, vol. 72, no. 2 (Feb. 1965), 36–46.

[Leino] Leino, K. R. M. *Puzzles*. research.microsoft.com/en-us/um/people/leino/puzzles.html (accessed Oct. 4, 2010).

[Lev06] Levitin, A. *Introduction to the Design and Analysis of Algorithms*, 2nd edition. Pearson, 2006.[3)]

[Lev81] Levmore, S. X., and Cook, E. E. *Super Strategies for Puzzles and Games*. Doubleday, 1981.

[Loy59] Loyd, S. (edited by M. Gardner) *Mathematical Puzzles of Sam Loyd*. Dover, 1959.[4)]

[Loy60] Loyd, S. (edited by M. Gardner) *More Mathematical Puzzles of Sam Loyd*. Dover, 1960.

[Luc83] Lucas, F. *Récréations mathématiques*, Vol. 2. Gauthier Villars, 1883.[5)]

[Mac92] Mack, D. R. *The Unofficial IEEE Brainbuster Gamebook: Mental Workouts for the Technically Inclined*. IEEE Press, 1992.

[Man89] Manber, U. *Introduction to Algorithms: A Creative Approach*. Addison-Wesley, 1989.

[1)] *Кордемский Б. А.* Математические завлекалки. — М.: Оникс, 2005.

[2)] *Курляндчик Л., Фомин Д.* Этюды о полуинварианте // Квант. — 1989. — № 7. — С. 63–68.

[3)] Русский перевод: *Левитин А.* Алгоритмы. Введение в разработку и анализ. — М.: Вильямс, 2006.

[4)] Русский перевод: *Лойд С.* Математическая мозаика. — М.: Мир, 1980.

[5)] Русский перевод: *Люка́ Ф.* Математические развлечения. — М.: Книжный клуб Книговек; СПб.: Северо-Запад, 2010.

[Mar96] Martin, G. E. *Polyominoes: A Guide to Puzzles and Problems in Tiling*. The Mathematical Association of America, 1996.

[MathCentral] *Math Central*. mathcentral.uregina.ca/mp (accessed Oct. 4, 2010).

[MathCircle] *The Math Circle* www.themathcircle.org/researchproblems.php (accessed Oct. 4, 2010).

[Mic09] Michael, T. S. *How to Guard an Art Gallery*. John Hopkins University Press, 2009.

[Mic08] Michalewicz, Z., and Michalewicz, M. *Puzzle-Based Learning: An Introduction To Critical Thinking, Mathematics, and Problem Solving*. Hybrid Publishers, 2008.

[Moo00] Moore, C., and Eppstein, D. One-dimensional peg solitaire and Duotaire. *Proceedings of MSRI Workshop on Combinatorial Games*, Berkeley, CA. MSRI Publications 42. Springer, 2000, 341–350.

[Mos01] Moscovich, I. 1000 *Play Thinks: Puzzles, Paradoxes, Illusions, and Games*. Workman Publishing, 2001.

[Nie01] Niederman, D. *Hard-to-Solve Math Puzzles*. Sterling Publishing, 2001.

[OBe65] O'Beirne, T. H. *Puzzles & Paradoxes*. Oxford University Press, 1965.

[Par95] Parberry, I. *Problems on Algorithms*. Prentice-Hall, 1995.

[Pet03] Peterson, I. Measuring with jugs. The Mathematical Association of America, June 2003. www.maa.org/mathland/mathtrek_06_02_03.html (accessed Oct. 4, 2010).

[Pet97] Petković, M. *Mathematics and Chess: 110 Entertaining Problems and Solutions*. Dover, 1997.

[Pet09] Petković, M. *Famous Puzzles of Great Mathematicians*. The American Mathematical Society, 2009.

[Pic02] Pickover, C. A. *The Zen of Magic Squares, Circles, and Stars: An Exhibition of Surprising Structures across Dimensions*. Princeton University Press, 2002.

[Poh72] Pohl, I. A sorting problem and its complexity. *Communications of the ACM*, vol. 15, issue 6 (June 1972), 462–464.

[Pol57] Pólya, G. *How to Solve It*: *A New Aspect of Mathematical Method*, 2nd ed. Princeton University Press, 1957.[1]

[Pou03] Poundstone, W. *How Would You Move Mount Fuji? Microsoft's Cult of the Puzzle—How the World's Smartest*

[1]Русский перевод: *Пойа Д.* Как решать задачу. — М.: Государственное учебно-педагогическое издательство министерства просвещения РСФСР, 1959.

Companies Select the Most Creative Thinkers. Little-Brown, 2003.[1)]

[Pre89] Pressman, I., and Singmaster, D. «The Jealous Husbands» and «The Missionaries and Cannibals.» *Mathematical Gazette*, 73, no. 464 (June 1989), 73–81.

[ProjEuler] *Project Euler*. projecteuler.net (accessed Oct. 2010).

[Ran09] Rand, M. On the Frame-Stewart algorithm for the Tower of Hanoi. www2.bc.edu/~grigsbyj/Rand_Final.pdf (accessed Oct. 4, 2010).

[Rob98] Robertson, J., and Webb, W. *Cake Cutting Algorithms*. A K Peters, 1998.

[Ros07] Rosen, K. *Discrete Mathematics and Its Applications*, 6th edition. McGraw-Hill, 2007.

[Ros38] Rosenbaum, J. Problem 319, *American Mathematical Monthly*, vol. 45, no. 10 (Dec. 1938), 694–696.

[Rot02] Rote, G. Crossing the bridge at night. *EATCS Bulletin*, vol. 78 (Aug. 2002), 241–246.

[Sav03] Savchev, S., and Andreescu, T. *Mathematical Miniatures*. The Mathematical Association of America, Anneli Lax New Mathematical Library, Volume #43, Washington, DC, 2003.

[Sch68] Schuh, F. *The Master Book of Mathematical Recreations*. Dover, 1968 (translated from Dutch).

[Sch04] Schumer, P. D. *Mathematical Journeys*. Wiley, 2004.

[Sch80] Schwartz, B. L., ed. *Mathematical Solitaires & Games*. (Excursions in Recreational Mathematics Series 1), Baywood Publishing, 1980.

[Sco44] Scorer, R. S., Grundy, P. M., and Smith, C. A. B. Some binary games. *Mathematical Gazette*, vol. 28, no. 280 (July 1944), 96–103.

[Sha02] Shasha, D. *Doctor Ecco's Cyberpuzzles*. Norton, 2002.

[Sha07] Shasha, D. *Puzzles for Programmers and Pros*. Wiley, 2007.

[Sillke] Sillke, T. Crossing the bridge in an hour. www.mathematik.uni-bielefeld.de/~sillke/PUZZLES/crossing-bridge (accessed Oct. 4, 2010).

[Sin10] Singmaster, D. *Sources in Recreational Mathematics: An Annotated Bibliography*, 8th preliminary edition. www.g4g4.com/MyCD5/SOURCES/SOURCE1.DOC (accessed Oct. 4, 2010).

[1)] Русский перевод: *Паундстоун У.* Как сдвинуть гору Фудзи? Подходы ведущих мировых компаний к поиску талантов. — М.: Альпина Паблишер; Харвест, 2008.

[Slo06] Slocum, J. and Sonneveld, D. *The 15 Puzzle: How It Drove the World Crazy. The Puzzle That Started the Craze of 1880. How America's Greatest Puzzle Designer, Sam Loyd, Fooled Everyone for 115 Years*. Slocum Puzzle Foundation, 2006.

[Sni02] Sniedovich, M. The bridge and torch problem. Feb. 2002. www.tutor.ms.unimelb.edu.au/bridge (accessed Oct. 4, 2010).

[Sni03] Sniedovich, M. OR/MS Games: 4. The Joy of Egg-Dropping in Braunschweig and Hong Kong. *INFORMS Transactions on Education*, vol. 4, no. 1 (Sept. 2003), 48–64.

[Spi02] Spivak, A. V. *One Thousand and One Mathematical Problems*. Education, 2002 (in Russian).[1)]

[Ste64] Steinhaus, H. *One Hundred Problems in Elementary Mathematics*. Basic Books, 1964 (translated from Polish).[2)]

[Ste04] Stewart, I. *Math Hysteria*. Oxford University Press, 2004.

[Ste06] Stewart, I. *How to Cut a Cake: And Other Mathematical Conundrums*. Oxford University Press, 2006.

[Ste09] Stewart, I. *Professor Stewart's Cabinet of Mathematical Curiosities*. Basic Books, 2009.[3)]

[Tan01] Tanton, J. *Solve This: Math Activities for Students and Clubs*. The Mathematical Association of America, 2001.

[techInt] *techInterviews*. www.techinterview.org/archive (accessed Oct. 4, 2010).

[Ton89] Tonojan, G. A. Canadian mathematical olympiads. *Kvant*, 1989, no. 7, 75–76 (in Russian).[4)]

[Tri69] Trigg, C. W. Inverting coin triangles. *Journal of Recreational Mathematics*, vol. 2 (1969), 150–152.

[Tri85] Trigg, C. W. *Mathematical Quickies*. Dover, 1985.[5)]

[Twe39] Tweedie, M. C. K. A graphical method of solving Tartaglian measuring puzzles. *Mathematical Gazette*, vol. 23, no. 255 (July 1939), 278–282.

---

1) *Спивак А. В.* Тысяча и одна задача по математике. 5–7 класс. — М.: Просвещение, 2002.

2) Русский перевод: *Штейнгауз Г.* Сто задач. — М.: Наука, 1976; 1982; 1986.

3) Русский перевод: *Стюарт И.* Математические диковинки профессора Стюарта. — М.: Лаборатория знаний, 2018. (Готовится к изданию. — *Прим. ред.*)

4) *Тонoян Г. А.* Канадские математические олимпиады // Квант. — 1989. — № 7. — С. 75–76.

5) Русский перевод: *Тригг Ч.* Задачи с изюминкой. — М.: Мир, 1975; 2000.

[Weiss] Weisstein, E. W. Josephus Problem. From *MathWorld*–A Wolfram Web Resource. mathworld.wolfram.com/JosephusProblem.html (accessed Oct. 4, 2010).

[Win04] Winkler, P. *Mathematical Puzzles: Connoisseur's Collection*. A K Peters, 2004.

[Win07] Winkler, P. *Mathematical Mind-Benders*. A K Peters, 2007.

[Zho08] Zhow, X. *A Practical Guide to Quantitative Finance Interview*. Lulu.com, 2008.

УКАЗАТЕЛЬ ГОЛОВОЛОМОК, СГРУППИРОВАННЫХ ПО МЕТОДАМ РАЗРАБОТКИ И АНАЛИЗА АЛГОРИТМОВ

В данном указателе головоломки сгруппированы по методам разработки и анализа алгоритмов. Головоломки, приведённые в учебном разделе, отмечены символами «УЧ», а всем остальным соответствуют их номера в основном разделе книги. Некоторые головоломки упомянуты в нескольких категориях.

АНАЛИЗ

Анализ выходных данных алгоритма

Подсчёт шагов алгоритма

Другое

ИНВАРИАНТЫ

Чётность

Раскраска

Другие инварианты

№ 15.	«*Замостить доску плитками тримино*»	делимость на 3
№ 40.	«*Перестановка четырёх коней*»	порядок по часовой стрелке
№ 92.	«*Замощение трапециями*»	делимость на 3
№ 96.	«*Замощение лестницы*»	делимость на 3
№ 104.	«*Разделение кучи фишек*»	сумма произведений
№ 105.	«*Головоломка „МИ“*»	делимость на 3
№ 110.	«*Хамелеоны*»	разности по модулю 3
№ 120.	«*Машина, распределяющая пенни*»	двоичное представление
№ 132.	«*Солдаты Конвея*»	вес позиции (полуинвариант)
№ 143.	«*Заражённая шахматная доска*»	периметр области (полуинвариант)
№ 149.	«*Распространение камушков*»	вес позиции

ПОИСК С ВОЗВРАТОМ

УЧ	«*Задача об n ферзях*»
№ 27.	«*Игра „Икосиан“*»
№ 29.	«*Ещё раз о магическом квадрате*»
№ 70.	«*Прыжки в пары — 1*»

УМЕНЬШАЙ И ВЛАСТВУЙ

Уменьшай на 1

УЧ	«*Задача о знаменитости*»	
№ 3.	«*Разделение прямоугольника*»	в обратном направлении
№ 4.	«*Отряд солдат*»	
№ 22.	«*Упорядочение списка команд*»	
№ 32.	«*Турнир на выбывание*»	
№ 33.	«*Магия и псевдомагия*»	в обратном направлении
№ 42.	«*Ещё раз о волке, козе и капусте*»	в обратном направлении
№ 43.	«*Расстановка чисел*»	
№ 46.	«*Фишки трёх цветов*»	
№ 59.	«*Шапки двух цветов*»	в обратном направлении
№ 65.	«*Угадывание кода*»	
№ 68.	«*Сумма цифр*»	
№ 81.	«*Ещё раз о задаче о знаменитости*»	

Уменьшай на 2

Уменьшай на другую постоянную величину

Уменьшай в постоянное число раз

№ 116. «*Подсчёт „пустых номеров“*» в 2 раза
№ 141. «*Задача Иосифа Флавия*» в 2 раза

Уменьшай на переменную величину

№ 23. «*Задача о польском национальном флаге*»
№ 28. «*Обвести фигуру*»
№ 84. «*Сортируем блины*»
№ 129. «*Головоломка Реве*»

РАЗДЕЛЯЙ И ВЛАСТВУЙ

УЧ «*Головоломка „Тримино“*»
№ 15. «*Замостить доску плитками тримино*»
№ 37. «*Задача о 2n шашках*»
№ 38. «*Замощение плитками тетрамино*»
№ 64. «*Восьмиугольники*»
№ 78. «*Замощение прямыми тримино*»
№ 80. «*Прогулка принца*»
№ 91. «*Горизонтальные и вертикальные домино*»
№ 92. «*Замощение трапециями*»
№ 95. «*Максимальный и минимальный вес*»
№ 96. «*Замощение лестницы*»
№ 101. «*Перекраска комнат*»
№ 113. «*Исчезающие монеты*»
№ 119. «*Замощение цветными тримино*»
№ 132. «*Солдаты Конвея*»

ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

УЧ «*Подсчёт кратчайших путей*»
№ 13. «*Преграждённые пути*»
№ 20. «*Спуск с максимальной суммой*»
№ 62. «*Робот собирает монетки*»
№ 98. «*Ромбопалиндром*»
№ 104. «*Разделение кучи фишек*»

ПОЛНЫЙ ПЕРЕБОР

УЧ «*Магический квадрат*»
№ 15. «*Замостить доску плитками тримино*»
№ 35. «*Три кувшина*»

ЖАДНЫЙ ПОДХОД

УЧ «*Неатакующие короли*»
УЧ «*Переход по мосту ночью*»
№ 24. «*Раскрашивание шахматной доски*»
№ 34. «*Монеты на звезде*»

УЧ	«*Головоломка Гуарини*»	граф, развёртывание графа
№ 9.	«*Счёт в уме*»	переупорядочение слагаемых
№ 11.	«*Столбик фальшивых монет*»	число монет соответствует номеру столбика
№ 25.	«*Лучшее время для жизни*»	интервалы на вещественной прямой
№ 34.	«*Монеты на звезде*»	граф, развёртывание графа
№ 40.	«*Перестановка четырёх коней*»	граф, развёртывание графа
№ 49.	«*Миссионеры и каннибалы*»	граф пространства состояний
№ 51.	«*Недостающее число*»	разность двух сумм; две последние цифры
№ 68.	«*Сумма цифр*»	переупорядочение слагаемых
№ 77.	«*Поиск закономерности*»	двоичная система
№ 106.	«*Лампочка и переключатели*»	битовая строка
№ 115.	«*Задача Баше о гирях*»	двоичная и троичная системы
№ 118.	«*Шесть коней*»	граф, развёртывание графа
№ 120.	«*Машина, распределяющая пенни*»	двоичная система
№ 125.	«*Отсортировать 5 за 7*»	точки на вещественной прямой
№ 130.	«*Отравленное вино*»	двоичная система и система по основанию 6
№ 135.	«*Разные пары*»	таблица; точки на окружности
№ 136.	«*Поимка шпиона*»	целочисленные пары
№ 146.	«*Стрельба по движущейся мишени*»	диаграмма
№ 148.	«*Свобода за одну монету*»	двоичная система

Сведение исходной задачи к другой

УЧ	«*Как оптимально разделить пирог*»	к математической задаче
№ 74.	«*Выбор места*»	к математической задаче

№ 89.	*«Перестановка фишек»*	к одномерной версии головоломки «Жабы и лягушки»
№ 92.	*«Замощение трапециями»*	к целому треугольнику
№ 102.	*«Обезьянка и кокосовые орехи»*	к математической задаче
№ 109.	*«Домино „дубль-n“»*	к задаче об эйлеровом цикле
№ 111.	*«Переворачивание треугольника из монеток»*	к математической задаче

ДРУГИЕ МЕТОДЫ

№ 1. *«Волк, коза и капуста»*
№ 7. *«Переход по мосту ночью»*
№ 12. *«Можно ли замостить доску?»*
№ 14. *«Переделать шахматную доску»*
№ 27. *«Игра „Икосиан“»*
№ 36. *«Ограниченное разнообразие»*
№ 38. *«Замощение плитками тетрамино»*
№ 41. *«Круг света»*
№ 44. *«Легче или тяжелее?»*
№ 58. *«Сортируем раз, сортируем два...»*
№ 60. *«Переделка треугольника из монеток в квадрат»*
№ 75. *«Инспекция бензоколонок»*
№ 88. *«Жабы и лягушки»*
№ 123. *«Задача о флаге Нидерландов»*
№ 137. *«Прыжки в пары — 2»*
№ 140. *«Снова задача об n ферзях»*
№ 142. *«Двенадцать монет»*
№ 147. *«Шапки с номерами»*

ПРЕДМЕТНО-ИМЕННОЙ УКАЗАТЕЛЬ

Минимальные системные требования определяются соответствующими требованиями программ Adobe Reader версии не ниже 11-й либо Adobe Digital Editions версии не ниже 4.5 для платформ Windows, Mac OS, Android и iOS; экран 10"

Научно-популярное электронное издание

Левитин Ананий, **Левитина** Мария

АЛГОРИТМИЧЕСКИЕ ГОЛОВОЛОМКИ

Ведущий редактор *М. С. Стригунова*. Художник *В. А. Прокудин*
Оригинал-макет подготовлен *О. Г. Лапко* в пакете LaTeX 2$_\varepsilon$

Подписано к использованию 07.02.19.
Формат 125×200 мм

Издательство «Лаборатория знаний»
125167, Москва, проезд Аэропорта, д. 3
Телефон: (499) 157-5272, e-mail: info@pilotLZ.ru,
http://www.pilotLZ.ru

Алгоритмические головоломки

МНОГИЕ СЧИТАЮТ, ЧТО АЛГОРИТМЫ ОТНОСЯТСЯ ИСКЛЮЧИТЕЛЬНО К ОБЛАСТИ ИНФОРМАТИКИ, НО СУТЬ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ – ИСПОЛЬЗОВАНИЕ АНАЛИТИЧЕСКОЙ ЛОГИКИ ДЛЯ РЕШЕНИЯ ЗАДАЧ.

Эта логика распространяется далеко за пределы информатики в обширный и интересный мир головоломок. В своей книге «Алгоритмические головоломки» Ананий и Мария Левитины показывают, как применять аналитическое мышление для решения головоломок с помощью строго определенных методов, используя в качестве примеров как классические головоломки, так и новые задачи, которые работодатели в больших компаниях предлагают решить во время собеседования при приеме на работу.

Уникальный сборник головоломок в этой книге дополнен учебным разделом по методам разработки и анализа алгоритмов, чтобы показать читателю различные подходы к решению алгоритмических задач. Для каждой из 150 головоломок в книге приводятся подсказка и решение, а также комментарии о происхождении головоломки и методах ее решения.

«Алгоритмические головоломки» – книга в своем роде уникальная, в ней есть головоломки разного уровня сложности. Читатели со средней математической подготовкой смогут развить навыки алгоритмического мышления с помощью головоломок элементарного уровня, а более подготовленные получат удовольствие от решения сложных головоломок.

АНАНИЙ ЛЕВИТИН – профессор информатики в Университете Вилланова. Он является автором популярного учебника по разработке и анализу алгоритмов, который был переведен на китайский, греческий, корейский и русский языки. Он также автор статей, посвященных математической теории оптимизации, разработке программного обеспечения, управлению данными, разработке алгоритмов и обучению информатике.

МАРИЯ ЛЕВИТИНА является независимым консультантом. Она проработала несколько лет в ведущих компаниях по разработке программного обеспечения и занималась прикладными бизнес-приложениями для больших корпораций; сейчас она специализируется на интернет-приложениях и беспроводных технологиях. Мария Левитина закончила Московский государственный университет по специальности «Математика».