

# Часть XIII

---

## ПОДСКАЗКИ

---

*Интервьюеры обычно не просто задают вам вопросы, ожидая, что вы ответите на них. Как правило, они помогают кандидатам подсказками в трудной ситуации, особенно в вопросах повышенной сложности. Полностью смоделировать процесс собеседования в книге невозможно, и все же эти подсказки помогут вам приблизиться к цели.*

*Постарайтесь решать задачи самостоятельно. Тем не менее, если задача окажется непосильной, не стесняйтесь обращаться за помощью. Затруднения — нормальная часть процесса собеседования.*

*Подсказки следуют в случайном порядке, чтобы подсказки к одной задаче не располагались по соседству. Это сделано для того, чтобы вы случайно не заглянули во вторую подсказку во время чтения первой.*

# 1

## Структуры данных

1	1.2	Что означает, что две строки являются перестановками друг друга? Присмотритесь к вашему определению. Можно ли проверить две строки по этому определению?
2	3.1	Стек — это просто структура данных, из которой первым извлекается последний добавленный элемент. Сможете ли вы смоделировать стек на базе массива? Помните, что возможных решений существует много, и у каждого из них есть свои плюсы и минусы
3	2.4	У этой задачи много решений, большинство из которых обладает одинаковой эффективностью по времени выполнения. Впрочем, код одних решений получается более коротким и простым. Сможете ли вы проанализировать разные решения?
4	4.10	Если T2 является поддеревом T1, как его порядок симметричного обхода связан с порядком симметричного обхода T1? А как насчет префиксного и постфиксного обхода?
5	2.6	Палиндром — строка, которая читается одинаково в прямом и обратном направлении. А если переставить в обратном порядке элементы связного списка?
6	4.12	Попробуйте упростить задачу. Что если путь должен начинаться от корня?
7	2.5	Конечно, связный список можно преобразовать в целые числа, вычислить сумму, а потом снова преобразовать ее в новый связный список. Если вы предложите этот вариант на собеседовании, скорее всего, интервьюер примет ответ, а потом предложит сделать то же без двойного преобразования
8	2.2	А если размер связного списка известен? Чем поиск K-го элемента с конца отличается от поиска X-го элемента?
9	2.1	Пробовали применить хеш-таблицу? Это можно сделать за один проход по связному списку
10	4.8	Если бы в каждом узле хранилась ссылка на его родителя, то можно было бы воспользоваться решением к задаче 2.7. Впрочем, интервьюер может отказать вам в этом предположении
11	4.10	Порядок симметричного обхода не принесет особой пользы. В конце концов, у всех бинарных деревьев поиска с одинаковыми значениями (независимо от структуры) этот порядок одинаков. (А если этот метод не работает в конкретном случае бинарного дерева поиска, то он не сработает и для более общего бинарного дерева.) С другой стороны, из порядка префиксного обхода можно извлечь намного больше полезной информации
12	3.1	Чтобы смоделировать хранение трех стеков в массиве, можно выделить первую треть массива под первый стек, вторую треть — под второй стек, и третью — под третий. Однако один стек может быть намного больше остальных. Нельзя ли организовать более гибкое разбиение массива?
13	2.6	Попробуйте воспользоваться стеком

14	4.12	Не забудьте, что пути могут перекрываться. Например, при поиске суммы 6 действителен как путь 1->3->2, так и путь 1->3->2->4->-6->2
15	3.5	Один из алгоритмов сортировки массива основан на переборе массива и вставке каждого элемента в новый массив в нужном порядке. Можно ли проделать то же со стеком?
16	4.8	Первым общим предком является узел с наибольшей глубиной, потомками которого являются $p$ и $q$ . Подумайте над тем, как идентифицировать этот узел
17	1.8	Если просто обнулять строку и столбец для каждого обнаруженного нуля, скорее всего, это кончится обнулением всей матрицы. Попробуйте найти ячейки с нулями до того, как вносить какие-либо изменения в матрицу
18	4.10	Возможно, вы решили, что если $T2.preorderTraversal()$ является подстрокой $T1.preorderTraversal()$ , то $T2$ является поддеревом $T1$ . Это почти правда... если деревья не содержат дубликатов. Допустим, $T1$ и $T2$ состоят только из дубликатов, но имеют разную структуру. Порядок префиксного обхода в этом случае будет совпадать, хотя $T2$ и не является поддеревом $T1$ . Как справиться с подобными ситуациями?
19	4.2	В минимальном бинарном дереве количество узлов в левом поддереве каждого узла примерно совпадает с количеством узлов в правом поддереве. Пока ограничимся корнем дерева. Как гарантировать, что в левом и правом поддереве корня количество узлов будет примерно одинаковым?
20	2.7	Это можно сделать за время $O(A+B)$ и с дополнительными затратами памяти $O(1)$ . Таким образом, можно обойтись и без хеш-таблицы (хотя возможно решение и с ней)
21	4.4	Задумайтесь над определением сбалансированного дерева. Возможно ли проверить это условие для одного узла? Возможно ли проверить его для всех узлов?
22	3.6	Можно рассмотреть возможность ведения одного связного списка для собак и кошек, а затем перебирать его содержимое до нахождения первой собаки (или кошки). К каким последствиям приведет такое решение?
23	1.5	Начнем с простого. Можно ли проверить каждое из условий по отдельности?
24	2.4	Задумайтесь над тем, что элементы не обязаны сохранять тот же относительный порядок. Необходимо лишь следить за тем, чтобы элементы, меньшие точки разбивки, предшествовали элементам, большим точки разбивки. Поможет ли это вам в поиске решений?
25	2.2	Если размер связного списка неизвестен, можно ли вычислить его? Как это отразится на времени выполнения?
26	4.7	Постройте направленный граф, представляющий зависимости. Каждый узел представляет проект, и ребро из $A$ в $B$ существует в том случае, если $B$ зависит от $A$ (проект $A$ должен быть построен до $B$ ). Также можно построить дерево наоборот, если вам так будет удобнее
27	3.2	Заметьте, что минимальный элемент изменяется достаточно редко. Он изменяется либо при добавлении меньшего элемента, либо при извлечении минимального элемента из стека
28	4.8	Как определить, является ли $p$ потомком узла $n$ ?
29	2.6	Допустим, длина связного списка известна. Сможете ли вы реализовать проверку с применением рекурсии?

30	2.5	Попробуйте применить рекурсию. Допустим, имеются два списка: $A = 1 \rightarrow 5 \rightarrow 9$ (представляет значение 951), $B = 2 \rightarrow 3 \rightarrow 6 \rightarrow 7$ (представляет значение 7632), и функция, работающая с усеченными списками ( $5 \rightarrow 9$ и $3 \rightarrow 6 \rightarrow 7$ ). Сможете ли вы воспользоваться ими для создания метода суммирования? Какими отношениями связаны вызовы $\text{sum}(1 \rightarrow 5 \rightarrow 9, 2 \rightarrow 3 \rightarrow 6 \rightarrow 7)$ и $\text{sum}(5 \rightarrow 9, 3 \rightarrow 6 \rightarrow 7)$ ?
31	4.10	Хотя на первый взгляд может показаться, что проблема возникает из-за дубликатов, на самом деле она глубже. Порядки префиксного обхода совпадают только из-за наличия пропущенных null-узлов. Рассмотрите возможность вставки условного значения в строку префиксного обхода для каждого null-узла. Регистрация null-узла как «реального» узла позволит различать разные структуры
32	3.5	Представьте, что вторичный стек отсортирован. Сможете ли вы вставить в него элементы в порядке сортировки? Возможно, для этого потребуется дополнительная память. Что можно использовать для хранения данных?
33	4.4	Если вы разработали решение методом «грубой силы», будьте осторожны с его временем выполнения. Если вы вычисляете высоту поддеревьев для каждого узла, алгоритм может получиться весьма неэффективным
34	1.9	Если строка получена в результате циклического сдвига другой строки, то циклический сдвиг выполняется в конкретной позиции. Например, циклический сдвиг строки «waterbottle» относительно 3-го символа означает, что строка «waterbottle» разделяется на 3-м символе, и правая половина («erbottle») размещается перед левой («wat»)
35	4.5	Если при симметричном обходе дерева элементы следовали в правильном порядке, означает ли это, что дерево действительно правильно упорядочено? Что произойдет при наличии дубликатов? Если дубликаты разрешены, они должны находиться с конкретной стороны (обычно слева)
36	4.8	Начните с корня. Можете ли вы определить, является ли корень первым общим предком? Если не является, то можете ли вы определить, с какой стороны корня находится первый общий предок?
37	4.10	Задачу также можно решать рекурсивно. Если задан конкретный узел в T1, можно ли проверить, совпадает ли его поддерево с T2?
38	3.1	Чтобы обеспечить гибкое распределение памяти, организуйте сдвиг стеков. Сможете ли вы обеспечить эффективное использование всего доступного пространства?
39	4.9	Каким должно быть самое первое значение, присутствующее в каждом массиве?
40	2.1	Без дополнительных затрат памяти потребуется время $O(N^2)$ . Попробуйте использовать два указателя, один из которых опережает другой
41	2.2	Попробуйте применить рекурсивную реализацию. Если вы можете найти (K-1)-й элемент с конца, удастся ли вам найти K-й элемент?
42	4.11	Будьте очень внимательны. Следите за тем, чтобы все узлы выбирались с равной вероятностью, а ваше решение не замедляло работу стандартных алгоритмов бинарных деревьев поиска (insert, find и delete). Также помните, что даже если дерево является сбалансированным бинарным деревом поиска, это не означает его полноты/законченности/идеальности
43	3.5	Поддерживайте вторичный стек в порядке сортировки, так что наибольшие элементы располагаются сверху. Используйте первичный стек для вспомогательного хранения
44	1.1	Попробуйте применить хеш-таблицу

45	2.7	Примеры упростят задачу. Нарисуйте диаграмму с пересекающимися связными списками и двумя эквивалентными (по значению) связными списками, которые не пересекаются
46	4.8	Примените рекурсивный подход. Проверьте, являются ли $p$ и $q$ потомками левого и правого поддерева. Если они являются потомками разных поддеревьев, то текущий узел является первым общим предком. Если они являются потомками одного поддерева, то это поддерево содержит первого общего предка. Как эффективно реализовать эту схему?
47	4.7	Взгляните на граф. Сможете ли вы найти на нем узел, который можно заведомо безопасно построить первым?
48	4.9	Корень — самое первое значение, которое должно присутствовать в каждом массиве. Что можно сказать о порядке значений в левом поддереве по сравнению со значениями в правом поддереве? Должны ли значения в левом поддереве вставляться до правого поддерева?
49	4.4	Нельзя ли изменить класс узла бинарного дерева, чтобы в нем хранилась высота его поддерева?
50	2.8	Задача на самом деле состоит из двух частей. (1) Определить, существует ли петля в связном списке. (2) Найти начало петли
51	1.7	Попробуйте действовать «по слоям». Возможно ли повернуть один конкретный слой?
52	4.12	Если каждый путь должен начинаться от корня, мы можем обойти все возможные пути, начиная от корня. Сумма отслеживается в процессе обхода, а счетчик <code>totalPaths</code> увеличивается при каждом нахождении пути с целевой суммой. Как распространить ситуацию на пути, который может начинаться с произвольного узла? Не забудьте: для начала стоит создать решение, работающее методом «грубой силы». Оптимизацией можно заняться позднее
53	1.3	Модификация строк часто упрощается, если начать с конца и продвигаться к началу
54	4.11	Вы сами создаете класс бинарного дерева поиска, поэтому можете включить в него любую информацию о структуре дерева или узлах (если только это не приведет к нежелательным последствиям — например, замедлению <code>insert</code> ). Вероятно, интервьюер именно поэтому указал, что класс пишется с нуля, а для эффективной реализации операции потребуется сохранить дополнительную информацию
55	2.7	Начните с простой проверки наличия пересечения
56	3.6	Предположим, для собак и кошек ведутся отдельные списки. Как найти самое старое животное каждого типа? Подойдите к делу творчески!
57	4.5	Чтобы дерево было бинарным деревом поиска, недостаточно выполнения условия <code>left.value &lt;= current.value &lt; right.value</code> для каждого узла. Любой узел в левом поддереве должен быть меньше текущего, а последний должен быть меньше любого узла в правом поддереве
58	3.1	Попробуйте рассматривать массив как циклический, то есть конец массива «переносится» в начало
59	3.2	А если в каждом элементе стека будут храниться дополнительные данные? Какие данные могут упростить решение задачи?
60	4.7	Если вам удастся найти узел, не имеющий входящих ребер, он определенно может быть построен. Найдите такой узел (их может быть несколько) и включите его в порядок построения. Что это будет означать для его исходящих ребер?

61	2.6	В рекурсивном решении (с известной длиной списка) середина является базовым случаем: значение <code>isPermutation(middle)</code> истинно. Возьмем узел $x$ непосредственно слева от середины: что можно сделать для проверки того, является ли $x \rightarrow \text{middle} \rightarrow y$ палиндромом? Теперь предположим, что проверка пройдена. Как насчет предшествующего узла $a$ ? Если $x \rightarrow \text{middle} \rightarrow y$ является палиндромом, как проверить, является ли палиндромом $a \rightarrow x \rightarrow \text{middle} \rightarrow y \rightarrow b$ ?
62	4.11	Можно ли воспользоваться алгоритмом обхода дерева для реализации этого алгоритма (как наивное решение методом «грубой силы»)? За какое время он будет выполняться?
63	3.6	Подумайте, как бы вы решили эту задачу в реальной жизни. Имеется список собак в хронологическом порядке и список кошек в хронологическом порядке. Какие данные необходимы для нахождения самого старого животного? Как организовать хранение этих данных?
64	3.3	Необходимо отслеживать размер каждого подстека. При заполнении одного стека можно создать новый стек
65	2.7	Два пересекающихся связанных списка всегда имеют одинаковый последний узел. После пересечения все последующие узлы совпадают
66	4.9	Отношения между значениями левого и правого поддеревя фактически произвольны. Значения левого поддерева могут быть вставлены до значений правого поддерева; возможно и обратное (правые значения предшествуют левым), как и любой другой порядок
67	2.2	В данном случае может быть полезно возвращать сразу несколько значений. В некоторых языках такая возможность не поддерживается напрямую, но обходные решения существуют в любом языке. Какие обходные решения такого рода вам известны?
68	4.12	Чтобы обобщить решение для путей с произвольным началом, достаточно повторить процесс для всех узлов
69	2.8	Чтобы проверить наличие петли, воспользуйтесь «методом бегунка» с двумя указателями, один из которых перемещается быстрее другого
70	4.8	В «наивном» алгоритме один метод проверял, является ли $x$ потомком $n$ , а другой метод осуществлял рекурсию для поиска первого общего предка. Такой подход приводит к повторению поиска по одним элементам поддерева. Поиск следует объединить в одну функцию <code>firstCommonAncestor</code> . Какие возвращаемые значения предоставят необходимую информацию?
71	2.5	Не забудьте выполнить проверку для связанных списков, имеющих разную длину
72	2.3	Представьте список $1 \rightarrow 5 \rightarrow 9 \rightarrow 12$ . После удаления элемента 9 он принимает вид $1 \rightarrow 5 \rightarrow 12$ . Доступен только узел 9. Сможете ли вы прийти к правильному ответу?
73	4.2	Реализация может быть основана на поиске следующего «идеального» элемента и многократного вызова <code>insertValue</code> . Однако такое решение будет немного неэффективным, поскольку оно требует повторного обхода дерева. Попробуйте воспользоваться рекурсией. Удастся ли вам разделить задачу на подзадачи?
74	1.8	Можно ли ограничиться затратами $O(N)$ дополнительного пространства вместо $O(N^2)$ ? Какая информация вам действительно необходима из списка нулевых ячеек?
75	4.11	Другое возможное решение — выбрать случайную глубину для обхода и выполнить случайный обход, остановившись при достижении выбранной глубины. Однако поразмыслите над этим решением. Будет ли оно работать?

76	2.7	Чтобы определить, пересекаются ли два связанных списка, можно перейти в конец каждого из них и сравнить конечные элементы
77	4.12	Если вы спроектировали алгоритм так, как описано выше, он выполняется за время $O(N \log N)$ в сбалансированном дереве. Это объясняется тем, что он содержит $N$ узлов, каждый из которых в худшем случае находится на глубине $O(\log N)$ . Узел обрабатывается один раз для каждого узла, расположенного выше него. Следовательно, $N$ узлов будут обработаны $O(\log N)$ раз. Существует оптимизация, которая обеспечит время выполнения $O(N)$
78	3.2	Допустим, каждый узел будет знать минимум своего «подстека» (всех элементов, расположенных под ним, включая его самого)
79	4.6	Подумайте над тем, как работает симметричный обход, и попробуйте «развернуть» его в обратном направлении
80	4.8	Функция <code>firstCommonAncestor</code> может вернуть первого общего предка (если $p$ и $q$ находятся в дереве); $p$ , если в дереве находится только $p$ , но не $q$ ; $q$ , если в дереве находится только $q$ , но не $p$ ; и <code>null</code> в остальных случаях
81	3.3	Извлечение элемента из конкретного подстека будет означать, что некоторые стеки не заполнены. Создаст ли это проблемы? Единственно правильного ответа не существует, но вы должны подумать над тем, как следует обрабатывать подобные ситуации
82	4.9	Разбейте задачу на подзадачи. Используйте рекурсию. Если известны все возможные последовательности для левого и правого поддеревьев, как создать все возможные последовательности для всего дерева?
83	2.8	Можно использовать два указателя, один из которых перемещается вдвое быстрее другого. Если в графе существует цикл, два указателя встретятся в одной позиции в одно время. Где именно они встретятся? Почему именно там?
84	1.2	Существует решение с временем $O(N \log N)$ . Другое решение использует дополнительную память, но обходится временем $O(N)$
85	4.7	Когда вы решаете построить узел, его исходящее ребро может быть удалено. Как после этого найти другие узлы, которые можно построить?
86	4.5	Если каждый узел в левом поддереве должен быть меньше текущего узла либо равен ему, это фактически означает, что самый большой узел в левом поддереве должен быть меньше текущего узла либо равен ему
87	4.12	Какая работа дублируется в текущем алгоритме «грубой силы»?
88	1.9	Фактически в задаче спрашивается, можно ли разбить первую строку на две части $x$ и $y$ , такие что первая строка имеет вид $xu$ , а вторая — $yx$ . Например, $x=wat$ , $y=erbottle$ . Первая строка $xu=waterbottle$ , а вторая $yx=erbottlewat$
89	4.11	Выбор случайной глубины особой пользы не принесет. Во-первых, на нижних уровнях больше узлов, чем на верхних. Во-вторых, даже если бы эти вероятности удалось сбалансировать, возможен «тупик», когда необходимо выбрать узел на глубине 5, а на глубине 3 обнаруживается лист. Впрочем, идея балансировки вероятностей выглядит интересно
90	2.8	Если вы еще не выявили закономерность, попробуйте взять связный список $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow ?$ , где $?$ — один из предшествующих узлов. Сделайте $?$ первым узлом (то есть $9$ содержит ссылку на $1$ , а весь связный список образует петлю). Затем сделайте $?$ узлом $2$ , затем узлом $3$ и $4$ . Какую закономерность вы наблюдаете? Можете ли вы объяснить, почему это происходит?

91	4.6	Один из шагов логики: «следующим» является крайний левый узел правого поддерева. А если правого поддерева нет?
92	1.6	Начните с простого. Проведите сжатие строки и сравните длины
93	2.7	Теперь нужно определить, где пересекаются связанные списки. Допустим, связанные списки имеют одинаковую длину. Как бы вы решили эту задачу?
94	4.12	Рассмотрите каждый путь, начинающийся от корня (существуют $N$ таких путей), как массив. По сути наш алгоритм «грубой силы» берет каждый массив и ищет в нем все смежные подпоследовательности, имеющие заданную сумму. Для этого мы вычисляем все подмассивы и их суммы. Возможно, стоит сосредоточиться на этой подзадаче. Как для заданного массива найти все смежные подпоследовательности с конкретной суммой? Еще раз поразмыслите над повторением работы в алгоритме «грубой силы»
95	2.5	Будет ли алгоритм работать для связанных списков вида $9 \rightarrow 7 \rightarrow 8$ и $6 \rightarrow 8 \rightarrow 5$ ? Перепроверьте этот момент
96	4.8	Будьте внимательны! А ваш алгоритм обрабатывает случай, в котором существует только один узел? Что произойдет? Возможно, возвращаемые значения придется слегка изменить
97	1.5	Как связаны между собой операции «вставка символа» и «удаление символа»? Нужны ли для них две разные проверки?
98	3.4	Главное различие между очередью и стеком — порядок элементов. Из очереди удаляется самый старый элемент, а из стека — самый новый. Как удалить из стека самый старый элемент, если доступен только самый новый?
99	4.11	Многие кандидаты предлагают наивное решение: выбрать случайное число от 1 до 3. Если выбрано число 1, то возвращается текущий узел; если 2 — происходит переход к левому поддереву, а если 3 — происходит переход к правому поддереву. Такое решение не работает. Почему? Можно ли исправить его так, чтобы оно заработало?
100	1.7	Поворот слоя означает перестановку значений в четырех массивах. Если бы вам предложили поменять местами значения двух массивов, как бы вы это сделали? Можно ли расширить решение для четырех массивов?
101	2.6	Вернитесь к предыдущей подсказке. Помните: существует возможность вернуть несколько значений. Для этого можно воспользоваться новым классом
102	1.8	Вероятно, вам понадобится память для хранения списка обнуляемых строк и столбцов. Удастся ли вам сократить дополнительные затраты памяти до $O(1)$ , используя для хранения данных саму матрицу?
103	4.12	Мы ищем подмассивы с суммой $targetSum$ . Заметим, что значение $runningSum_i$ (сумма элементов от 0 до $i$ ) может быть получено за постоянное время. Чтобы подмассив элементов от $i$ до $j$ имел сумму $targetSum$ , значение $runningSum_{i-1} + targetSum$ должно быть равно $runningSum_j$ (попробуйте нарисовать элементы массива или цепочку чисел). Так как мы можем отслеживать $runningSum$ «на ходу», как быстро найти количество индексов $i$ , для которых выполняется это уравнение?
104	1.9	Поразмыслите над предыдущей подсказкой. Затем подумайте, что произойдет, если объединить строку <code>erbottlewat</code> с ней самой. Вы получите <code>erbottlewaterbottlewat</code>
105	4.4	Для хранения высоты поддерева не нужно изменять класс бинарного дерева. Сможет ли ваша рекурсивная функция вычислить высоту каждого поддерева одновременно с проверкой сбалансированности узла? Попробуйте написать функцию, которая возвращает сразу несколько значений
106	1.4	Генерировать все перестановки не обязательно. Более того, так поступать не рекомендуется — это будет крайне неэффективно



107	4.3	Попробуйте изменить алгоритм поиска в графе, чтобы он отслеживал глубину от корня
108	4.12	Попробуйте использовать хеш-таблицу, связывающую значение <code>runningSum</code> с количеством элементов, обладающих этим значением <code>runningSum</code>
109	2.5	К дополнительному вопросу: проблема в том, что при разной длине связанных списков начало одного связанного списка может представлять тысячи, а начало другого — десятки. А если привести списки к одинаковой длине? Возможно ли изменить связанный список без изменения представляемого им значения?
110	1.6	Примите меры к тому, чтобы избежать многократной конкатенации строк. Такая конкатенация была бы очень неэффективной
111	2.7	Если два связанных списка имеют одинаковую длину, вы можете перемещаться по каждому из них, пока не найдете общий элемент. Какие изменения следует внести в алгоритм, если списки имеют разную длину?
112	4.11	Предыдущее решение (с выбором случайного числа от 1 до 3) не работает из-за того, что вероятности узлов не равны. Например, корень будет возвращаться с вероятностью $1/3$ , даже если дерево содержит более 50 узлов. Очевидно, все узлы не могут иметь вероятность $1/3$ . Проблему можно решить выбором случайного числа от 1 до размера дерева. Впрочем, проблема будет решена только для корня. А как насчет остальных узлов?
113	4.5	Вместо того чтобы проверять значение текущего узла по значениям <code>leftTree.max</code> и <code>rightTree.min</code> , нельзя ли действовать наоборот — проверить узлы левого дерева и убедиться в том, что они меньше <code>current.value</code> ?
114	3.4	Чтобы удалить из стека самый старый элемент, можно многократно удалить новейший элемент (и вставить его во временный стек), пока в стеке не останется один элемент. Затем после извлечения все элементы возвращаются на место. Проблема в том, что несколько извлечений из стека подряд потребуют времени $O(N)$ . Сможете ли вы предложить оптимизацию для нескольких извлечений подряд?
115	4.12	После того как вы проработаете алгоритм для поиска в массиве всех смежных подмассивов с заданной суммой, попробуйте применить его к дереву. Помните, что в процессе обхода и модификации хеш-таблицы может возникнуть необходимость в «восстановлении» ее содержимого при возврате
116	4.2	Допустим, у нас имеется метод <code>createMinimalTree</code> , возвращающий минимальное дерево для заданного массива (который по какой-то странной причине не работает с корнем). Сможете ли вы воспользоваться им для решения задачи с корнем? Сможете ли написать базовый случай для функции? Отлично! Фактически вы сделали все, что нужно
117	1.1	Нельзя ли воспользоваться битовым вектором?
118	1.3	Возможно, вам понадобится узнать количество пробелов. Нельзя ли просто подсчитать их?
119	4.11	Проблема предыдущего решения заключается в том, что на одной стороне поддерева может быть больше узлов, чем на другой. Следовательно, необходимо взвесить вероятности выбора левого или правого поддерева в зависимости от количества узлов с каждой из сторон. Как будет работать такое решение? Как узнать количество узлов?
120	2.7	Попробуйте использовать разность между длинами двух связанных списков
121	1.4	Какими характеристиками должна обладать строка, которая является перестановкой палиндрома?
122	1.2	Может, воспользоваться хеш-таблицей?

123	4.3	Также может пригодиться хеш-таблица или массив, связывающий номер уровня с количеством узлов на этом уровне
124	4.4	Возможно написать одну функцию <code>checkHeight</code> , которая совмещает вычисление высоты с проверкой сбалансированности. Целочисленное возвращаемое значение может обозначать оба условия
125	4.7	Попробуйте применить совершенно иной подход: поиск в глубину от произвольного узла. Как связаны между собой поиск в глубину и допустимый порядок построения проектов?
126	2.2	Возможно ли итеративное решение? Представьте, что у вас имеются два указателя, установленные на соседние узлы, которые перемещаются с одинаковой скоростью по связанному списку. Когда один указатель достигнет конца связанного списка, где будет находиться другой?
127	4.1	Для решения этой задачи существуют два хорошо известных алгоритма. Какими достоинствами и недостатками обладает каждый из них?
128	4.5	Рассмотрите <code>checkBST</code> как рекурсивную функцию, которая проверяет, что каждый узел находится в пределах допустимого диапазона (минимум, максимум). Сначала этот диапазон бесконечен. При обходе левого поддерева минимумом становится отрицательная бесконечность, а максимумом — <code>root.value</code> . Сможете ли вы реализовать эту рекурсивную функцию и обеспечить необходимую корректировку диапазонов в процессе обхода дерева?
129	2.7	Если переместить указатель вперед по более длинному связанному списку на разность длин, вы сможете применить подход, аналогичный ситуации с равной длиной связанных списков
130	1.5	Можно ли выполнить все три проверки за один проход?
131	1.2	Две строки перестановок должны состоять из одинаковых символов, но следующие в разном порядке. Можно ли привести их к одному порядку?
132	1.1	Возможно ли решить задачу за время $O(N \log N)$ ? Как будет выглядеть такое решение?
133	4.7	Выберите произвольный узел и проведите от него поиск в глубину. Добравшись до конца пути, мы знаем, что этот узел может быть последним, потому что никакие другие узлы от него не зависят. Что это значит для узлов, непосредственно предшествующих ему?
134	1.4	Вы пытались воспользоваться хеш-таблицей? Это позволит сократить время до $O(N)$
135	4.3	Попробуйте разработать алгоритм, сочетающий поиск в глубину с поиском в ширину
136	1.4	Нельзя ли сократить затраты памяти за счет применения битового вектора?

# 2

## Концепции и алгоритмы

137	5.1	Разбейте формулировку на части. Начните со сброса соответствующих битов
138	8.9	Попробуйте применить метод «Базовый случай с расширением»
139	6.9	Если взять конкретную дверь $x$ , на каких проходах будет изменяться ее состояние?
140	11.5	Что имеет в виду интервьюер? Существует много разных типов ручек. Составьте список вопросов для уточнения информации
141	7.11	Задача не настолько сложна, как кажется на первый взгляд. Начните с составления списка ключевых объектов в системе, а затем подумайте над тем, как они взаимодействуют друг с другом
142	9.6	Для начала сформулируйте исходные предположения. Что нужно и что не нужно строить в этой задаче?
143	5.2	Чтобы разобраться в сути задачи, подумайте, как бы вы решили ее для целых чисел
144	8.6	Попробуйте применить метод «Базовый случай с расширением»
145	5.7	Перестановка каждой пары означает, что четные биты перемещаются налево, а нечетные направо. Сможете ли вы разбить задачу на части?
146	6.10	Сможете ли вы разделить бутылки на группы? Помните, что тестовую полосу нельзя повторно использовать после получения положительного результата, но при отрицательном результате она может использоваться сколько угодно раз
147	5.4	Начните с решения методом «грубой силы» для каждого случая
148	8.14	Можно ли просто проверить все возможные варианты? На что это будет похоже?
149	6.5	Поэкспериментируйте с сосудами и переливаниями. Удастся ли вам отмерить любой объем, кроме 3 и 5 л?
150	8.7	Допустим, у вас имеются все перестановки $abc$ . Как использовать их для получения всех перестановок $abcd$ ?
151	5.5	Проанализируйте задачу, начиная с внешнего слоя и заканчивая внутренним
152	8.1	Начните с конца. На сколько ступенек ребенок переместился на последнем шаге?
153	7.1	Понятие «колода карт» достаточно широкое. Попробуйте сформулировать разумные ограничения для задачи
154	6.7	Заметьте, что в каждой семье только одна девочка
155	8.13	А если ящики отсортировать, это упростит задачу?
156	6.8	Это алгоритмическая задача, и к ней следует подходить соответственно. Предложите решение методом «грубой силы», вычислите количество бросков в худшем случае, а затем попытайтесь оптимизировать его

157	6.9	При каком условии столкновений не будет?
158	9.6	Предполагается, что остальной код системы уже написан, и нам остается лишь разобраться со статистикой продаж. Значит, мы можем каким-то образом получать уведомления о новых покупках
159	5.3	Начните с решения методом «грубой силы». Можно ли проверить все возможные варианты?
160	6.7	Попробуйте записать потомство каждой семьи в виде последовательности символов B (мальчик) и G (девочка)
161	8.8	Проблему можно было бы решить простой проверкой дубликатов перед выводом (или добавлением в список). Это можно сделать при помощи хеш-таблицы. В каких ситуациях такое решение возможно, а когда неприемлемо?
162	9.7	Будет ли это приложение интенсивно выполнять чтение или запись данных?
163	6.10	Существует относительно простое решение, которое выполняется за 28 дней в худшем случае. Впрочем, есть и более эффективные решения
164	11.5	Допустим, ручка предназначена для детей. Что это может означать? Какие возможны сценарии использования?
165	9.8	Четко определите масштаб задачи. Что будет, а что не будет рассматриваться как часть системы?
166	8.5	Рассмотрите умножение 8 на 9 как подсчет количества ячеек в матрице с шириной 8 и высотой 9
167	5.2	Что означает каждая цифра в числе 0,893 (в десятичной записи)? Что тогда означает каждая цифра в числе вида 0,10010 (в двоичной записи)?
168	8.14	Каждый вариант может рассматриваться как позиция для вставки круглых скобок. В нашем случае выражение разбивается по операторам. Что считать базовым случаем?
169	5.1	Для сброса битов создайте маску, которая выглядит как серия 1, за которой следует серия 0, и снова серия 1
170	8.3	Начните с алгоритма «грубой силы»
171	6.7	Задачу можно попытаться решить на математическом уровне, хотя вычисления будут довольно трудными. Возможно, вам будет проще дать оценку для семей, включающих, скажем, до 6 детей. Хорошего математического доказательства вы не получите, но зато поймете, в каком направлении следует искать ответ
172	6.9	В каком случае дверь останется открытой в конце процесса?
173	5.2	Число 0,893 (в десятичной записи) означает $8 * 10^{-1} + 8 * 10^{-2} + 3 * 10^{-3}$ . Адаптируйте эту систему к основанию 2
174	8.9	Допустим, нам известны все действительные способы записи двух пар круглых скобок. Как использовать эту информацию для получения действительных способов записи трех пар скобок?
175	5.4	Представьте двоичное число как последовательность 0 и 1. Допустим, 0 заменяется на 1, а 1 заменяется на 0. В каком случае число увеличится? В каком случае оно уменьшится?
176	9.6	Подумайте над тем, какие ожидания предъявляются к актуальности и точности данных. Должны ли данные всегда быть актуальными на 100%? Может, для каких-то продуктов точность данных важнее, чем для других?
177	10.2	Как узнать, являются ли два слова анаграммами? Подумайте над определением «анаграммы». Объясните его своими словами

178	8.1	Если нам известно количество путей для каждой из ступенек до 100-й, можно ли вычислить количество путей к 100-й ступеньке?
179	7.8	Стоит ли использовать для представления белых и черных фишек один класс? Каковы достоинства и недостатки такого решения?
180	9.7	Заметьте, что в систему будет поступать большое количество данных, но, скорее всего, читаться эти данные будут относительно редко
181	6.2	Вычислите вероятности победы в первой и второй игре, а затем сравните их
182	10.2	Два слова являются анаграммами, если они состоят из одинаковых символов, следующих в разном порядке. Как упорядочить символы?
183	6.10	Почему между тестированием и получением результатов существует такая задержка? Задача не зря сформулирована именно так, а не «Свести к минимуму количество тестов»
184	9.8	Насколько равномерно, по вашему мнению, распределяется трафик? Будут ли все документы получать примерно одинаковую долю трафика? Или некоторые документы могут оказаться особенно популярными?
185	8.7	Метод 1: перестановки строки $abc$ представляют все способы упорядочения $abc$ . Допустим, теперь нужно создать все варианты упорядочения $abcd$ . Возьмем конкретное упорядочение, например $bdca$ . Строка $bdca$ также представляет вариант упорядочения $abc$ : удалив $d$ , вы получите $bca$ . Можно ли на основе заданной строки $bca$ создать все «связанные» упорядочения, которые также включают $d$ ?
186	6.1	Весы разрешено использовать только один раз. Это означает, что должны использоваться все (или почти все) баночки. При этом с ними необходимо обращаться по-разному, иначе различить их не удастся
187	8.9	Можно попытаться сгенерировать решение для трех пар, взяв список для двух пар скобок и добавив третью пару. Третья пара может добавляться до, после или в середине списка (то есть $()\langle\text{РЕШЕНИЕ}\rangle$ , $(\langle\text{РЕШЕНИЕ}\rangle)$ , $\langle\text{РЕШЕНИЕ}\rangle()$ ). Сработает ли такой способ?
188	6.7	Логический подход может оказаться проще математического. Представьте, что все рождения записываются в огромную строку из $B$ и $G$ . Обратите внимание: группировка по семьям для данной задачи несущественна. Какова вероятность того, что следующим символом, добавленным в строку, будет $B$ , а не $G$ ?
189	9.6	Покупки будут происходить очень часто. Вероятно, частоту записи в базу данных следует ограничить
190	8.8	Если вы еще не решили задачу 9.7, начните с нее
191	6.10	Нельзя ли одновременно выполнять несколько тестов?
192	7.6	Стандартный прием решения таких головоломок — разделение фрагментов на боковые и внутренние. Как представить это различие на объектно-ориентированном уровне?
193	10.9	Начните с наивного решения (но не слишком наивного — используйте тот факт, что матрица отсортирована)
194	8.13	Ящики можно отсортировать по убыванию любого размера. Так будет получено частичное упорядочение: ящики, стоящие ближе к концу массива, должны предшествовать ящикам, стоящим ближе к началу
195	6.4	Муравьи не столкнутся только в том случае, если все они идут в одном направлении. Какова вероятность того, что все три муравья идут по часовой стрелке?

196	10.11	Представьте, что массив отсортирован по возрастанию. Можно ли «подправить» порядок элементов так, чтобы массив состоял из чередующихся пиков и впадин?
197	8.14	Базовый случай – одно значение (1 или 0)
198	7.3	Начните с определения масштаба задачи и составьте список предположений. Часто в разумных предположениях нет ничего плохого, но вы должны сформулировать их явно
199	9.7	Для системы будет характерна высокая интенсивность записи: данные импортируются в большом объеме, но редко читаются
200	8.7	Метод 1: чтобы для заданной строки (например, $bcac$ ) создать все перестановки $abcd$ , у которых $\{a, b, c\}$ следуют в порядке $bca$ , можно вставить $d$ во все возможные позиции: $dbcac, bdca, bcda, bcad$ . Если известны все перестановки $abc$ , сможете ли вы сгенерировать все перестановки $abcd$ ?
201	6.7	Биология не изменилась: изменились только условия, при которых семья перестает заводить детей. При каждой беременности с вероятностью 50% родится мальчик, и с вероятностью 50% родится девочка
202	5.5	Что означает $A \& B == \emptyset$ ?
203	8.5	Чтобы подсчитать ячейки в матрице $8 \times 9$ , можно подсчитать ячейки в матрице $4 \times 9$ и удвоить результат
204	8.3	Вероятно, ваш алгоритм «грубой силы» выполняется за время $O(N)$ . Если вы попытаетесь улучшить этот результат, какого времени вы рассчитываете добиться? Для каких алгоритмов характерно такое время?
205	6.10	Решение 2: попробуйте вычислить номер бутылки, цифру за цифрой. Как узнать первую цифру номера отравленной бутылки? А как насчет второй цифры? Третьей?
206	9.8	Как вы собираетесь генерировать URL-адреса?
207	10.6	Сравните сортировку слиянием с быстрой сортировкой. Какой из двух алгоритмов лучше подойдет в данном случае?
208	9.6	Постарайтесь ограничить количество соединений, это очень затратная операция
209	8.9	У решения из предыдущей подсказки есть один недостаток: оно может содержать дубликаты. От дубликатов можно избавиться при помощи хеш-таблицы
210	11.6	Будьте осторожны с предположениями. Кто ваши пользователи? Где они будут использовать продукт? На первый взгляд ответы очевидны, но простота может оказаться обманчивой
211	10.9	Можно провести бинарный поиск по каждой строке. Сколько времени займет такое решение? Можно ли его улучшить?
212	9.7	Подумайте над тем, как вы будете получать банковские данные (будут ли они загружаться в активном или пассивном режиме?), какие возможности должны поддерживаться системой, и т. д.
213	7.7	Как обычно, определите масштаб задачи. Являются ли «дружеские» отношения взаимными? Существуют ли статусные сообщения? Будете ли вы поддерживать групповой чат?
214	8.13	Попробуйте разбить задачу на подзадачи
215	5.1	Создать маску с 0 в начале или в конце несложно. Но как создать маску с нулями в середине? Есть простой способ: создайте маски для левой и правой части и объедините их
216	7.11	Какими отношениями связаны файлы и каталоги?

217	8.1	Количество путей до 100 можно вычислить по количеству путей до 99, 98 и 97 (путь завершается прыжком на соответственно 1, 2 или 3 ступеньки). Что делать с этими числами — складывать или умножать? А именно $f(100) = f(99) + f(98) + f(97)$ или $f(100) = f(99) * f(98) * f(97)$ ?
218	6.6	Задача решается на логическом уровне, а не на уровне хитроумной игры слов. Примените логику/математику/алгоритмы для ее решения
219	10.11	Попробуйте перебрать отсортированный массив. Нельзя ли менять элементы местами, пока не будет получен нужный результат?
220	11.5	Вы рассмотрели все варианты использования — как целевые (рисование и т. д.), так и нецелевые? Как насчет безопасности? Ручка, предназначенная для детей, должна быть безопасной
221	6.10	Решение 2: будьте очень внимательны с граничными случаями. Что если третья цифра в номере совпадает со второй или первой цифрой?
222	8.8	Попробуйте подсчитать количество вхождений каждого символа. Например, строка ABCAAC содержит 3 вхождения A, 2 вхождения C и 1 вхождение B
223	9.6	Не забудьте, что продукт может относиться к нескольким категориям
224	8.6	Наименьший диск можно легко переложить с одной башни на другую. Также легко перекладываются два диска. Как переложить три наименьших диска?
225	11.6	В реальном собеседовании также стоило бы обсудить, какие средства тестирования доступны в данной ситуации
226	5.3	Переход 0 в 1 может привести к слиянию двух последовательностей 1, но только в том случае, если эти последовательности разделены всего одним нулем
227	8.5	Подумайте, что делать с нечетными числами
228	7.8	Какой класс должен вести счет игры?
229	10.9	Если вы рассматриваете конкретный столбец, существует ли способ быстро исключить его из поиска (по крайней мере в некоторых случаях)?
230	6.10	Решение 2: дополнительный день тестирования можно потратить на проверку цифры 3 другим способом. И снова будьте осторожны с граничными случаями
231	10.11	Если пики находятся на своих местах, то и впадины тоже будут расставлены правильно. Следовательно, при переборе можно пропускать каждый второй элемент
232	9.8	Если URL-адреса будут генерироваться случайным образом, нужно ли беспокоиться о коллизиях (два документа с одинаковыми URL)? Если нужно, то как решать эту проблему?
233	6.8	Для начала можно применить некое подобие бинарного поиска: начинаем с 50-го этажа, потом бросаем с 75-го, 88-го и т. д. Проблема в том, что если при броске с 50-го этажа яйцо разобьется, то второе яйцо придется бросать, начиная с 1-го этажа и продвигаясь на 1. В худшем случае потребуется 50 попыток (50-й этаж, 1-й этаж, 2-й этаж и так далее до 49-го). Можно ли улучшить этот результат?
234	8.5	Если часть работы повторяется между рекурсивными вызовами, можно ли кэшировать результаты?
235	10.7	Попробуйте применить битовый вектор
236	9.6	Где стоит кэшировать данные или ставить задачи в очередь?
237	8.1	Значения умножаются в ситуации «сначала делаем это, потом это». Значения складываются в ситуации «делаем либо это, либо это»

238	7.6	Как бы вы сохранили позицию найденного фрагмента? Возможно ли хранение по строке и столбцу?
239	6.2	Чтобы вычислить вероятность победы во второй игре, начните с вычисления вероятности попадания при первом и втором броске с промахом на третьем
240	8.3	Можно ли решить задачу за время $O(\log N)$ ?
241	6.10	Решение 3: рассматривайте каждую тестовую полосу как бинарный индикатор «яд есть/яда нет»
242	5.4	Если перевести 1 в 0, а 0 в 1, число увеличится только в том случае, если замена 0->1 произошла в более старшем разряде, чем замена 1->0. Как использовать это обстоятельство для создания следующего большего числа (с тем же количеством единиц)?
243	8.9	Также можно рассмотреть возможность перемещения по строке и добавления левой или правой скобки на каждом шаге. Приведет ли это к устранению дубликатов? Как узнать, возможно ли добавить левую или правую скобку?
244	9.6	Возможно, в зависимости от сделанных предположений удастся обойтись даже без базы данных. Что это будет означать для системы? Насколько хороша эта мысль?
245	7.7	В этой задаче следует подумать об основных компонентах системы и технологиях, которые могут пригодиться при проектировании
246	8.5	При вычислении результата $9 \times 7$ (оба числа нечетные) можно вычислить $4 \times 7$ и $5 \times 7$ , а затем сложить результаты
247	9.7	Попробуйте исключить избыточные запросы к базе данных. Если в долгосрочном хранении данных нет необходимости, возможно, база данных вообще не понадобится
248	5.7	Можно ли создать число, представляющее только четные биты? А потом сдвинуть четные биты на 1?
249	6.10	Решение 3: если полоска является бинарным индикатором, нельзя ли связать целочисленные ключи с набором из 10 бинарных индикаторов, чтобы каждый ключ имел уникальное представление?
250	8.6	Попробуйте переместить меньший диск с башни $X=0$ на башню $Y=2$ с использованием башни $Z=1$ как временного хранилища для решения $f(1, X=0, Y=2, Z=1)$ . Перемещение двух наименьших дисков записывается в виде $f(2, X=0, Y=2, Z=1)$ . У вас есть решение для $f(1, X=0, Y=2, Z=1)$ и $f(2, X=0, Y=2, Z=1)$ ; сможете ли вы привести решение для $f(3, X=0, Y=2, Z=1)$ ?
251	10.9	Так как каждый столбец отсортирован, вы знаете, что искомое значение не может содержаться в столбце, если оно меньше минимального значения в этом столбце. Что еще из этого следует?
252	6.1	А если положить на весы по одной таблетке из каждой баночки? А если по две таблетки?
253	10.11	Насколько необходима сортировка массивов? Можно ли решить задачу для несортированного массива?
254	10.7	Можно ли выполнить несколько проходов для сокращения затрат памяти?
255	8.8	Чтобы получить все перестановки с тремя символами $A$ , двумя символами $B$ и одним символом $C$ , сначала необходимо выбрать начальный символ: $A$ , $B$ или $C$ . Если это $A$ , то необходимо знать все перестановки с двумя-тремя символами $A$ , двумя символами $B$ и одним символом $C$
256	10.5	Попробуйте применить бинарный поиск



257	11.1	Код содержит две ошибки
258	7.4	Сколько уровней на стоянке — один, несколько? Какие «возможности» она должна поддерживать? Является ли автостоянка платной? Какие виды транспортных средств на ней могут размещаться?
259	9.5	Вероятно, вам придется сделать некоторые предположения (отчасти потому, что вы не сидите перед интервьюером). Это нормально, но ваши предположения должны быть явно сформулированы
260	8.13	Подумайте, какое первое решение вам придется принять. Первое решение — какой ящик должен находиться внизу штабеля
261	5.5	Если $A \& B == 0$ , это означает, что A и B никогда не содержат 1 в одном разряде. Примените этот факт к решаемому уравнению
262	8.1	Каким временем выполнения характеризуется этот метод? Хорошенько подумайте. Можно ли его улучшить?
263	10.2	Нельзя ли воспользоваться стандартным алгоритмом сортировки?
264	6.9	Если целое число $x$ делится на $a$ , $a \mid b = x / a$ , то $x$ также делится на $b$ . Означает ли это, что все числа имеют четное количество множителей?
265	8.9	Добавление левой или правой круглой скобки на каждом шаге приведет к устранению дубликатов. Каждая подстрока будет уникальной на каждом шаге. Следовательно, вся строка будет уникальной
266	10.9	Если значение $x$ меньше начала столбца, то оно также не может находиться в столбцах, находящихся справа
267	8.7	Чтобы сгенерировать все перестановки $abcd$ , вычислите все перестановки $abc$ и вставьте $d$ в каждую возможную позицию в этих перестановках
268	11.6	Какие различные возможности и варианты использования следует протестировать?
269	5.2	Как получить первую цифру в дробной части 0,893? Если умножить значение на 10, оно станет равным 8,93. Что произойдет, если умножить его на 2?
270	9.2	Какой вид поиска лучше подойдет для поиска соединения между узлами — в ширину или в глубину? Почему?
271	7.7	Как узнать, что пользователь отключился от сервера?
272	8.6	Заметим, что не так важно, какая башня является источником, приемником или буфером. Чтобы выполнить $f(3, X=0, Y=2, Z=1)$ , можно сначала выполнить $f(2, X=0, Y=1, Z=2)$ (переместить два диска с башни 0 на башню 1, используя башню 2 в качестве буфера), затем переместить диск 3 с башни 0 на башню 2, а потом $f(2, X=1, Y=2, Z=0)$ (переместить два диска с башни 1 на башню 2, используя башню 0 в качестве буфера). Как происходит повторение этого процесса?
273	8.4	Как построить все подмножества $\{a, b, c\}$ на базе подмножеств $\{a, b\}$ ?
274	9.5	Подумайте, как бы вы проектировали решение для одной машины. Может, использовать хеш-таблицу? Как будет работать эта схема?
275	7.1	Как вы собираетесь обрабатывать тузы?
276	9.7	Как можно большая часть работы должна выполняться асинхронно
277	10.11	Предположим, имеется последовательность из трех элементов $\{0, 1, 2\}$ в произвольном порядке. Запишите все возможные последовательности этих элементов. Как изменить их, чтобы сделать элемент 1 пиком?
278	8.7	Метод 2: можно ли сгенерировать все перестановки подстрок из трех символов, располагая всеми перестановками символов из двух строк?
279	10.9	Рассмотрите предыдущую подсказку в контексте строк

280	8.5	Другой способ: при подсчете ячеек в матрице $9 \times 7$ можно вычислить результат $4 \times 7$ , удвоить его, а затем прибавить 7
281	10.7	Попробуйте использовать один проход для получения диапазона значений, а второй — для выбора конкретного числа
282	6.6	Предположим, на острове ровно один голубоглазый житель. Что он увидит? Когда покинет остров?
283	7.6	Для каких фрагментов проще всего подобрать соответствие? Можно ли начать с них? Какие фрагменты станут самыми простыми после этого?
284	6.2	Если два события являются взаимоисключающими (то есть никогда не происходят одновременно), их вероятности складываются
285	9.2	Вероятно, лучше использовать поиск в ширину. Поиск в глубину может идти по длинному пути даже в том случае, если кратчайший путь совсем короткий. Возможна ли модификация поиска в ширину, которая работает еще быстрее?
286	8.3	Бинарный поиск выполняется за время $O(\log N)$ . Можно ли применить разновидность бинарного поиска в этой задаче?
287	7.12	Для разрешения коллизий хеш-таблица должна представлять собой массив связанных списков
288	10.9	Что произойдет, если информация будет храниться в массиве? Какими достоинствами и недостатками обладает это решение?
289	10.8	Можно ли воспользоваться битовым вектором?
290	8.4	Все, что является подмножеством $\{a, b\}$ , также является подмножеством $\{a, b, c\}$ . Какие множества являются подмножествами $\{a, b, c\}$ , но не $\{a, b\}$ ?
291	10.9	Можно ли использовать предыдущие подсказки для перемещения вверх, вниз, вправо и влево по строкам и столбцам?
292	10.11	Вернитесь к только что записанному набору последовательностей для $\{\emptyset, 1, 2\}$ . Представьте, что перед крайним левым элементом находятся другие элементы. Вы уверены, что ваш способ перестановки не приведет к тому, что предыдущая часть массива станет недействительной?
293	9.5	Можно ли объединить хеш-таблицу со связным списком, чтобы использовать преимущества обеих структур данных?
294	6.8	Лучше немного уменьшить высоту первого броска. Например, можно сначала сбросить яйцо с 10-го этажа, потом с 20-го, 30-го и т. д. В худшем случае это потребует 19 бросков (10, 20, ..., 100, 91, 92, ..., 99). Можно ли улучшить это значение? Не пытайтесь изобретать разные решения, думайте глубже. Как определяется худший случай? Как в нем учитывается количество бросков каждого яйца?
295	8.9	Проверка строки может осуществляться простым подсчетом левых и правых круглых скобок. Добавление левой скобки всегда возможно вплоть до общего количества пар. Добавление правой скобки возможно при условии <i>количество левых скобок <math>\leq</math> количество правых скобок</i>
296	6.4	Ситуацию можно рассматривать так: искомая вероятность равна сумме вероятностей того, что все 3 муравья идут по часовой стрелке, и того, что все 3 муравья идут против часовой стрелки. А можно взглянуть на происходящее иначе: первый муравей выбирает направление. Какова вероятность того, что два других муравья выберут то же направление?
297	5.2	Подумайте, что случится со значениями, не имеющими точного представления в двоичном виде
298	10.3	Можно ли изменить алгоритм бинарного поиска для этой цели?

299	11.1	Что произойдет с <code>unsigned int</code> ?
300	8.11	Попробуйте разбить задачу на подзадачи. Если бы вы набирали сдачу монетами, то какое первое решение вы бы приняли?
301	10.10	К недостаткам массивов относится медленная вставка. Какие еще структуры данных можно использовать?
302	5.5	Если $(n \& (n - 1)) == 0$ , это означает, что $n$ и $n - 1$ не содержат единиц в одном и том же разряде. Почему?
303	10.9	Представьте прямоугольник, нарисованный от ячейки в направлении правого нижнего угла матрицы. Значение ячейки будет больше всех элементов в прямоугольнике
304	9.2	Можно ли вести поиск как от начального, так и от конечного узла? По какой причине или в каком случае он будет работать быстрее?
305	8.14	Если у вас получился длинный код с большим количеством команд <code>if</code> (для каждого возможного оператора проверяется логический результат и левая/правая сторона), подумайте о связи между разными частями. Попробуйте упростить код и повторно использовать готовые результаты
306	6.9	Число 3 имеет четное количество множителей (1 и 3). Число 12 имеет четное количество множителей (1, 2, 3, 4, 6, 12). Какие числа не обладают этим свойством? Что это говорит о состоянии дверей?
307	7.12	Тщательно продумайте информацию, которая должна храниться в узле связанного списка
308	8.12	Мы знаем, что в каждом ряду доски должен стоять ферзь. Можно ли перебрать все возможные комбинации?
309	8.7	Метод 2: чтобы сгенерировать перестановку <code>abcd</code> , необходимо выбрать исходный символ. Возможны четыре варианта: <code>a</code> , <code>b</code> , <code>c</code> и <code>d</code> . Затем генерируются перестановки трех оставшихся символов. Как использовать этот метод для генерирования перестановок всей строки?
310	10.3	Каким временем выполнения обладает ваш алгоритм? Что произойдет, если массив может содержать дубликаты?
311	9.5	Как масштабировать это решение при увеличении системы?
312	5.4	Можно ли изменить 0 на 1 для создания следующего большего числа?
313	11.4	Подумайте, что должны проверять нагрузочные тесты. Какие факторы задействованы в нагрузке веб-страницы? По каким критериям будет оцениваться работоспособность веб-страницы под высокой нагрузкой?
314	5.3	Длину каждой последовательности можно увеличить, объединив ее со смежной последовательностью (если она имеется) или просто изменяя состояние соседнего нуля. Остается найти лучший вариант
315	10.8	Рассмотрите возможность самостоятельной реализации класса битового вектора — это хорошее упражнение является важной частью данной задачи
316	10.11	Ваша цель — разработать алгоритм с временем выполнения $O(n)$
317	10.9	Значение в ячейке должно быть больше всех элементов, находящихся ниже и правее. Оно должно быть меньше всех элементов, находящихся выше и левее. Если бы мы захотели сначала исключить максимальное количество элементов, то с каким элементом следовало бы сравнивать значение $x$ ?

318	8.6	Если у вас возникают проблемы с рекурсией, попробуйте больше доверять рекурсивному процессу. Разобравшись с тем, как переместить два верхних диска с башни 0 на башню 2, считайте, что это перемещение просто работает. Когда потребуется переместить три диска, не задумывайтесь над тем, как именно это делается. Два диска переместились. Что теперь делать с третьим?
319	6.1	Представьте, что баночек всего три, и одна содержит более тяжелые таблетки. Допустим, вы положили на весы разное количество таблеток из каждой баночки (например, 5 из первой, 2 из второй и 9 из третьей). Что покажут весы?
320	10.4	Задумайтесь над тем, как работает бинарный поиск. Какие проблемы могут возникнуть с обычной реализацией бинарного поиска?
321	9.2	Как бы вы могли реализовать эти алгоритмы и эту систему в реальном мире? Какие оптимизации вы бы при этом применили?
322	8.13	После того как будет выбран нижний ящик, необходимо выбрать второй ящик. Потом третий и т. д.
323	6.2	Вероятность двух и более попаданий из трех бросков равна сумме: вероятность(попадание при броске 1, попадание при броске 2, промах при броске 3) + вероятность(попадание при броске 1, промах при броске 2, попадание при броске 3) + вероятность(промах при броске 1, попадание при броске 2, попадание при броске 3) + вероятность(попадание при броске 1, попадание при броске 2, попадание при броске 3)
324	8.11	Если бы вы набирали сдачу, то, скорее всего, начали бы с монет в 25 центов
325	11.2	Примите во внимание как внутренние, так и внешние факторы (за пределами системы)
326	9.4	Оцените, сколько памяти для этого понадобится
327	8.14	Проанализируйте рекурсию. Встречаются ли в ней повторные вызовы? Можно ли применить мемоизацию?
328	5.7	Значение 1010 в двоичной записи соответствует 10 в десятичной записи или 0xA в шестнадцатеричной записи. Как будет выглядеть в шестнадцатеричной записи последовательность 101010...? Иначе говоря, как бы вы представили последовательность из чередующихся 1 и 0 (1 в нечетных позициях)? А если единицы стоят в четных позициях?
329	11.3	Рассмотрите как граничные, так и более общие случаи
330	10.9	Сравнив $x$ с центральным элементом матрицы, можно исключить приблизительно четверть элементов
331	8.2	Чтобы робот достиг последней ячейки, он должен сначала добраться до предпоследних ячеек, а чтобы добраться до предпоследних ячеек, нужно найти путь ко вторым ячейкам с конца
332	10.1	Попробуйте двигаться от конца массива к началу
333	6.8	Если ронять яйцо 1 с фиксированными интервалами (например, через каждые 10 этажей), то худший случай будет равен сумме худших случаев для яйца 1 и яйца 2. Недостаток предыдущих решений заключается в том, что с увеличением объема работы для яйца 1 объем работы яйца 2 не уменьшался. В идеале ситуацию хотелось бы сбалансировать: с увеличением объема работы яйца 1 (которое переживает большее количество бросков) на долю яйца 2 должно оставаться меньше работы. Что это может означать?
334	9.3	Подумайте о том, как могут возникнуть бесконечные циклы

335	8.7	Метод 2: чтобы сгенерировать все перестановки $abcd$ , выберите каждый символ ( $a$ , $b$ , $c$ и $d$ ) в качестве исходного. Сгенерируйте останки остальных символов и поставьте в начало исходный символ. Как генерировать перестановки остальных символов? При помощи рекурсивного процесса, следующего той же логике
336	5.6	Как вычислить количество различающихся битов в двух числах?
337	10.4	Бинарный поиск основан на сравнении со средним элементом. Чтобы найти средний элемент, необходимо знать длину. В нашем случае длина неизвестна. Можно ли ее узнать?
338	8.4	Подмножествами, содержащими $c$ , могут быть только подмножества $\{a, b, c\}$ , но не подмножества $\{a, b\}$ . Сможете ли вы построить эти подмножества из подмножеств $\{a, b\}$ ?
339	5.4	Переход из 0 в 1 создает большее число. Чем ближе позиция перехода к младшим разрядам, тем меньше будет это большее число. В числе вида 1001 нужно изменить крайний правый нуль (получится число 1011). С другой стороны, в числе 1010 изменять крайнюю правую единицу не следует
340	8.3	Возможно ли для заданного индекса и значения определить, где будет находиться «волшебный» индекс — до или после него?
341	6.6	Теперь предположим, что на острове живут двое голубоглазых. Что они увидят? Какие смогут сделать выводы? Когда покинут остров? Вспомните свой ответ из предыдущей подсказки. Считайте, что этот ответ им известен
342	10.2	А нужна ли полноценная «сортировка»? Нельзя ли обойтись простой реорганизацией списка?
343	8.11	После того как вы решили использовать две монеты по 25 центов для представления суммы в 98 центов, остается решить, как набрать 48 центов монетами по 10, 5 и 1 центу
344	7.5	Продумайте всю функциональность, которая должна поддерживаться системой онлайн-библиотеки. Не старайтесь учесть все до последней мелочи, но проследите за тем, чтобы все предположения были явно сформулированы
345	11.4	А может, построить собственные инструменты? Как они могут выглядеть?
346	5.5	Каким отношением связаны $n$ и $n - 1$ ? Проанализируйте процесс вычитания в двоичной записи
347	9.4	Для решения задачи потребуется несколько проходов? Несколько машин?
348	10.4	Длину можно определить проверкой с экспоненциальным приращением: сначала проверяется индекс 2, затем индекс 4, затем 8, 16 и т. д. Каким временем выполнения будет характеризоваться этот алгоритм?
349	11.6	Что можно автоматизировать?
350	8.12	В каждом ряду должен стоять ферзь. Начнем с последнего ряда. Есть восемь позиций, в которых может стоять ферзь. Можно ли опробовать их все?
351	7.10	Должны ли ячейки с числами, пустые ячейки и ячейки с минами быть представлены разными классами?
352	5.3	Попробуйте сделать это за линейное время, с одним проходом и затратами памяти $O(1)$
353	9.3	Как обнаружить уже посещавшуюся страницу? Что это будет означать для работы алгоритма?
354	8.4	Остальные подмножества можно построить включением $c$ во все подмножества $\{a, b\}$

355	5.7	Попробуйте использовать маски 0хаааааааа и 0х55555555 для выбора нечетных и четных битов, а затем примените сдвиг к четным и нечетным битам для создания нужного числа
356	8.7	Метод 2: в реализации этого метода рекурсивная функция может возвращать список строк, к которому присоединяется начальный символ
357	6.8	Попробуйте бросать яйцо 1 с большими интервалами вначале, которые затем постоянно уменьшаются. Идея заключается в том, чтобы сумма бросков яиц 1 и 2 оставалась по возможности постоянной. Для каждого дополнительного броска яйца 1 количество бросков яйца 2 уменьшается на 1. Как выбрать правильный интервал?
358	5.4	Следует изменять состояние крайнего правого незавершающего 0. Число 1010 превратится в 1110. После этого остается переключить 1 в 0, чтобы число стало по возможности меньшим, но превышающим исходное число (1010). Как следует действовать? Как уменьшить число?
359	8.1	Попробуйте применить мемоизацию для оптимизации неэффективной рекурсивной программы
360	8.2	Сначала немного упростите задачу, определив, существует ли путь. Затем измените алгоритм для отслеживания пути
361	7.10	Как выглядит алгоритм расстановки мин на игровом поле?
362	11.1	Присмотритесь к параметрам printf
363	7.2	Прежде чем браться за написание кода, составьте список необходимых объектов и основных алгоритмов. Есть ли у вас все необходимое для работы?
364	8.10	Представьте формулировку задачи в виде графа
365	9.3	Как определить, совпадают ли две страницы? По URL? По содержанию? Оба варианта могут оказаться не идеальными. Почему?
366	5.8	Сначала опробуйте наивный подход. Можно ли установить конкретный пиксел?
367	6.3	Представьте кости домино, разложенные на шахматной доске. Сколько черных квадратов они будут закрывать? Сколько белых квадратов?
368	8.13	После того как базовый рекурсивный алгоритм будет реализован, подумайте, как оптимизировать его. Удастся ли вам найти повторяющиеся подзадачи?
369	5.6	Вспомните, как работает операция XOR. Если имеется выражение $a \text{ XOR } b$ , где в результате находятся 1? Где находятся 0?
370	6.6	Продолжайте развивать эту схему. А если на острове живут трое голубоглазых? Четверо?
371	8.12	Выделите несколько меньших подзадач. Ферзь в ряду 8 должен находиться в столбце 1, 2, 3, 4, 5, 6, 7 или 8. Можно ли вывести все способы размещения восьми ферзей, в которых ферзь находится в ряду 8 и столбце 3? Далее необходимо проверить все способы размещения ферзя в ряду 7
372	5.5	При выполнении двоичного вычитания крайний правый 0 последовательно заменяется на 1 до тех пор, пока мы не доберемся до разряда 1 (который также меняет свое состояние). Все остальные разряды (с 1 и 0), находящиеся слева, остаются в неизменном состоянии
373	8.4	Для этого можно связать каждое подмножество с двоичным числом. $i$ -й бит числа представляет «флаг» вхождения элемента в множество

374	6.8	Пусть $X$ — общее количество бросков. Это означает, что яйцо 2 должно сделать $X - 1$ бросков, если яйцо 1 разобьется. Сумма бросков яиц 1 и 2 должна быть по возможности постоянной. Если яйцо 1 разобьется со второго броска, то яйцо 2 должно сделать $X - 2$ броска, и т. д. Чему равно $X$ ?
375	5.4	Чтобы уменьшить число, нужно переместить все 1 как можно дальше справа от измененного бита
376	10.10	Насколько хорошо сработает бинарное дерево поиска?
377	7.10	Как случайно разместить мины на поле? Вспомните алгоритм тасования колоды карт. Можно ли применить аналогичный метод?
378	8.13	Также можно рассматривать происходящее как последовательность решений: первый ящик включается в штабель? Второй ящик включается в штабель? И так далее
379	6.5	Если заполнить 5-литровый кувшин, а потом из него заполнить 3-литровый кувшин, в 5-литровой кувшине останется 2 литра. Далее можно либо оставить эти два литра, либо опустошить меньший кувшин и перелить два литра в него
380	8.11	Проанализируйте алгоритм. Присутствует ли в нем повторяющаяся работа? Можно ли оптимизировать процесс?
381	5.8	Когда вы рисуете длинную линию, целые байты превращаются в последовательности единиц. Можно ли установить их за одну операцию?
382	8.10	Реализация может быть основана на поиске в глубину (или на поиске в ширину). Каждый соседний пиксел «правильного» цвета представляется соединенным узлом
383	5.5	Представьте числа $n$ и $n - 1$ . Чтобы вычесть 1 из $n$ , следует перевести крайнюю правую 1 в 0, а все 0 справа от него — в 1. Если $n \& n - 1 = 0$ , то справа от первой единицы других единиц нет. Что это означает относительно $n$ ?
384	5.8	Как насчет начала и конца линии? Придется ли устанавливать эти пиксели по отдельности или же их можно установить сразу?
385	9.1	Представьте, что вы проектируете реальное приложение. Какие факторы при этом следует учесть?
386	7.10	Как подсчитать количество мин рядом с ячейкой? Нужно ли перебирать все ячейки?
387	6.1	Нужно найти формулу, которая позволит вычислить баночку с тяжелыми таблетками в зависимости от веса
388	8.2	Еще раз подумайте об эффективности алгоритма. Можно ли оптимизировать его?
389	7.9	Метод <code>rotate()</code> должен выполняться за время $O(1)$
390	5.4	После того как вы решите задачу получения следующего числа, попробуйте обратить логику для получения предыдущего
391	5.8	Правильно ли ваш код обрабатывает случай, при котором $x_1$ и $x_2$ находятся в одном байте?
392	10.10	Рассмотрите дерево бинарного поиска, в котором в каждом узле хранятся дополнительные данные
393	11.6	А вы подумали о безопасности и надежности?
394	8.11	Попробуйте применить мемоизацию
395	6.8	У меня в худшем случае получилось 14 бросков. А у вас?
396	9.1	Единственного правильного ответа не существует. Рассмотрите несколько вариантов технической реализации

397	6.3	Сколько черных клеток на доске? Сколько белых?
398	5.5	Мы знаем, что если $n \& (n - 1) == 0$ , $n$ содержит только одну единицу. Какие числа содержат только одну единицу?
399	7.10	Если пользователь щелкает на пустой ячейке, по какому алгоритму будет происходить расширение соседних ячеек?
400	6.5	После того как будет разработано решение задачи, рассмотрите задачу в более широкой перспективе. Если имеются два кувшина объемом $X$ и $Y$ литров, всегда ли с их помощью можно отмерить объем $Z$ ?
401	11.3	Возможно ли протестировать все возможные ситуации? Как выбрать приоритетные направления тестирования?



# 3

## Вопросы, основанные на знаниях

402	12.9	Сначала сосредоточьтесь на концепциях, а затем переходите к конкретной реализации. Как должен выглядеть класс SmartPointer?
403	15.2	Переключение контекста представляет собой переключение между двумя процессами. Это происходит тогда, когда выполнение одного процесса прекращается для передачи управления другому процессу
404	13.1	Задумайтесь над тем, для кого доступны приватные методы
405	15.1	Чем они отличаются в отношении использования памяти?
406	12.11	Вспомните, что двумерный массив по сути является массивом массивов
407	15.2	В идеале нам хотелось бы зафиксировать промежуток времени между «остановкой» одного процесса и «запуском» другого. Но как определить, когда происходит переключение?
408	14.1	Условие GROUP BY может вам помочь
409	13.2	Когда выполняется блок finally? Существуют ли ситуации, в которых он выполнен не будет?
410	12.2	Можно ли выполнить операцию «на месте»?
411	14.2	Возможно, решение будет полезно разбить на две части. Первая часть получает идентификатор здания и количество открытых запросов, а вторая получает название здания
412	13.3	Учтите, что смысл некоторых конструкций может изменяться в зависимости от того, где они применяются
413	12.10	Обычно вызов malloc просто возвращает произвольный блок памяти. Если это поведение невозможно изменить, можно ли поработать с результатом и получить то, что требуется?
414	15.7	Сначала реализуйте однопоточное решение FizzBuzz
415	15.2	Попробуйте создать два процесса и организовать передачу небольшого блока данных между ними. Для этого системе придется остановить один процесс и активизировать другой
416	13.4	Их цели похожи, но чем различаются реализации?
417	15.5	Как гарантировать, что вызов first() будет завершен до вызова second()?
418	12.11	Одно из возможных решений — вызвать malloc для каждого массива. Как освободить память в этом случае?
419	15.3	Взаимная блокировка возникает при появлении цикла в порядке ожидания. Как разорвать этот цикл?
420	13.5	Подумайте над тем, какие структуры данных следует использовать
421	12.7	Подумайте, для чего используются виртуальные методы
422	15.4	Если каждый процесс будет заранее объявлять, какие процессы ему могут понадобиться, можно ли обнаружить потенциальные взаимные блокировки заранее?

423	12.3	Какие структуры данных задействованы в каждом из двух случаев? К каким последствиям это приводит?
424	13.5	HashMap использует массив связанных списков. TreeMap использует красно-черное дерево. LinkedHashMap использует двусвязные блоки. Что из этого следует?
425	13.4	Например, чем они отличаются в отношении использования примитивных типов?
426	12.11	Можно ли выделить память таким образом вместо одного смежного блока?
427	12.8	Эту структуру данных можно представить в виде бинарного дерева, но это не обязательно. Что если в структуре присутствует цикл?
428	14.7	Вероятно, понадобятся списки студентов, учебных курсов и еще одна таблица, определяющая отношения между студентами и курсами. Заметьте, что эти отношения относятся к типу «многие-ко-многим»
429	15.6	Ключевое слово <code>synchronized</code> гарантирует, что два потока не смогут одновременно выполнять синхронизированные методы для одного экземпляра
430	13.5	Подумайте, чем они могут отличаться в порядке перебора ключей. Почему один вариант может оказаться предпочтительнее другого?
431	14.3	Для начала попытайтесь получить список идентификаторов (только идентификаторов) всех квартир, относящихся к запросу
432	12.10	Допустим, имеется последовательность целых чисел (3, 4, 5, ...). Какая длина последовательности гарантирует, что одно из чисел будет кратно 16?
433	15.5	Почему в данном случае не следует использовать флаги <code>boolean</code> ?
434	15.4	Подумайте над порядком запросов в графе. Как будет выглядеть взаимная блокировка в таком графе?
435	13.6	Рефлексия позволяет получить информацию о методах и полях объектов. Для чего может понадобиться эта информация?
436	14.6	Будьте особенно внимательны с типами отношений: один-к-одному, один-ко-многим, многие-ко-многим
437	15.3	Например, можно запретить философу брать палочку, если он не может взять другую
438	12.9	Подумайте над отслеживанием количества ссылок
439	15.7	Не пытайтесь проявить «творческий подход» в однопоточном решении. Просто напишите простой и удобочитаемый код
440	12.10	Как освободить память?
441	15.2	Если ваше решение не идеально, это абсолютно нормально. Возможно, идеал просто недостижим. Обсудите достоинства и недостатки вашего решения
442	14.7	Тщательно подумайте над тем, как будут обрабатываться совпадения при выборе верхних 10%
443	13.8	Наивное решение — выбор случайного подмножества размером $z$ с последующим перебором элементов и включением их в множество с вероятностью $z/\text{размер\_списка}$ . Почему такое решение не работает?
444	14.5	Денормализация подразумевает включение избыточных данных в таблицу. Обычно она применяется в очень крупных системах. Какую пользу может она принести?

445	12.5	При поверхностном копировании копируется только исходная структура данных. При глубоком копировании делается это, а также копируются данные, на которые указывают ссылки. Для каких ситуаций лучше подойдет каждый из этих вариантов?
446	15.5	Может, воспользоваться семафорами?
447	15.7	Опишите общую схему многопоточного решения. На синхронизацию пока не обращайте внимания
448	13.7	Сначала подумайте, как бы вы реализовали решение без лямбда-выражений
449	12.1	Как бы вы это сделали, если бы количество строк в файле уже было известно?
450	13.8	В списке всех подмножеств $n$ -элементного множества для любого заданного элемента $x$ одна половина подмножеств содержит $x$ , а другая не содержит
451	14.4	Опишите особенности внутренних (INNER) и внешних (OUTER) соединений. Внешние соединения делятся на несколько типов: левые, правые, полные
452	12.2	Будьте осторожны с нуль-символом
453	12.9	Какие методы/операторы стоило бы переопределить?
454	13.5	За какое время будут выполняться стандартные операторы?
455	14.5	Подумайте о высокозатратных соединениях в большой системе
456	12.6	Ключевое слово <code>volatile</code> означает, что переменная может измениться под влиянием внешних факторов (например, другого процесса). Почему это может быть необходимо?
457	13.8	Не выбирайте длину подмножества заранее — это не обязательно. Рассматривайте ситуацию как последовательность решений о том, входит ли каждый элемент в множество или нет
458	15.7	Когда вы разберетесь со структурой всех потоков, подумайте, что необходимо сделать для синхронизации
459	12.1	Предположим, количество строк в файле заранее не известно. Можно ли решить задачу без предварительного подсчета строк?
460	12.7	Что произошло бы, если бы деструктор не был виртуальным?
461	13.7	Разбейте задачу на две части: фильтрацию списка стран и получение суммы
462	12.8	Попробуйте воспользоваться хеш-таблицей
463	12.4	Вспомните о таблицах виртуальных функций
464	13.7	Как это сделать без фильтрации?

# 4

## Дополнительные задачи

465	16.3	Формулировку задачи стоит уточнить
466	16.12	Рассмотрите возможность применения рекурсии или дерева
467	17.1	Смоделируйте двоичное сложение вручную (шаг за шагом!) и постарайтесь понять, что же при этом происходит
468	16.13	Нарисуйте квадрат и линии, разделяющие его пополам. Где находятся все эти линии?
469	17.24	Начните с решения методом «грубой силы»
470	17.14	Возможных решений несколько. Попробуйте найти их. Начать можно с наивного решения
471	16.20	Попробуйте использовать рекурсию
472	16.3	Все ли линии пересекаются? Что определяет наличие пересечения?
473	16.7	Пусть $k = 1$ , если $a > b$ , и $0$ в противном случае. Если значение $k$ известно, сможете ли вы вернуть максимум (без сравнений и логики <code>if-else</code> )?
474	16.22	Сложности возникают с бесконечной сеткой. Какие возможности вы предложите?
475	17.15	Попробуйте упростить задачу: а если нужно узнать самое длинное слово, состоящее из двух слов в списке?
476	16.10	Можно ли подсчитать количество людей, живущих в каждый год?
477	17.25	Начните с группировки словаря по длине слов, так как известно, что все столбцы должны иметь одинаковую длину (как и строки)
478	17.7	Начните с наивного решения: объединения имен, которые являются синонимами. Как обнаружить отношения транзитивности? Если $A == B$ , $A == C$ и $C == D$ , из этого следует, что $A == D == B == C$
479	16.13	Каждая линия, разделяющая квадрат надвое, проходит через центр квадрата. Как найти линию, которая разделяет два квадрата?
480	17.17	Начните с решения методом «грубой силы». Каким временем выполнения оно характеризуется?
481	16.22	Вариант 1: действительно ли необходима бесконечная сетка? Перечитайте задачу. Известен ли максимальный размер сетки?
482	16.16	Поможет ли информация о самых длинных отсортированных последовательностях в начале и конце?
483	17.2	Попробуйте решить задачу с применением рекурсии
484	17.26	Начните с простого алгоритма, сравнивающего все документы со всеми остальными документами. Как быстрее всего вычислить степень сходства двух документов?
485	17.5	На самом деле неважно, какая перед вами буква или цифра. Можно упростить задачу до массива с символами $A$ и $B$ . Тогда от вас потребуется найти самый длинный подмассив с равным количеством $A$ и $B$

486	17.11	Сначала рассмотрите алгоритм нахождения минимального расстояния для однократного выполнения. Задача должна решаться за время $O(N)$ , где $N$ — количество слов в документе
487	16.20	Можно ли рекурсивно опробовать все возможности?
488	17.9	Разберитесь, что же от вас требуется в задаче. По сути вам предлагается найти $k$ -е наименьшее число вида $3^a * 5^b * 7^c$
489	16.2	Подумайте, каким должно быть лучшее возможное время в этой задаче. Если ваше решение совпадает с лучшим возможным временем, то, скорее всего, улучшить его уже не удастся
490	16.10	Попробуйте воспользоваться хеш-таблицей или массивом, связывающим год рождения с количеством людей, которые жили в этот год
491	16.14	Иногда метод «грубой силы» оказывается неплохим решением. Можно ли перебрать все возможные строки?
492	16.1	Попробуйте представить два числа $a$ и $b$ на числовой оси
493	17.7	В этой задаче центральное место занимает группировка синонимов. После того как это будет сделано, вычислить частоты будет несложно
494	17.3	Решите задачу 17.2, если вы еще не сделали этого ранее
495	17.16	У задачи есть два решения: рекурсивное и итеративное. Наверное, проще начать с рекурсивного решения
496	17.13	Опробуйте рекурсивное решение
497	16.3	Бесконечные линии почти всегда пересекаются, если только они не параллельны. Впрочем, даже параллельные линии могут «пересекаться», если они совпадают. Что это означает для отрезков?
498	17.26	Чтобы вычислить степень сходства двух документов, попробуйте провести реорганизацию данных. Отсортировать? Использовать другую структуру данных?
499	17.15	Если бы мы захотели узнать самое длинное слово, состоящее из других слов списка, можно было бы перебрать все слова по убыванию длины и проверить, возможно ли составить их из других слов. Для этого строку нужно будет разбить во всех возможных позициях
500	17.25	Можно ли найти прямоугольник с заданной длиной и шириной? А если просто перебрать все возможные варианты?
501	17.11	Адаптируйте свой «одноразовый» алгоритм для многократного применения. Какая часть будет самой медленной? Можно ли ее оптимизировать?
502	16.8	Рассматривайте число как набор фрагментов из трех цифр
503	17.19	Начните с первой части: поиска недостающего числа, если отсутствует только одно число
504	17.16	Рекурсивное решение: для каждого сеанса возможны два варианта (включить сеанс в расписание или не включать). Решение методом «грубой силы» может перебрать все возможные комбинации. Однако следует обратить внимание, что при включении заявки $i$ рекурсивный алгоритм должен пропустить заявку $i + 1$
505	16.23	Будьте особенно внимательны с тем, чтобы решение возвращало все значения от 0 до 6 с одинаковой вероятностью
506	17.22	Начните с рекурсивного решения методом «грубой силы». Просто создайте все слова, находящиеся на расстоянии одного изменения, проверьте, присутствуют ли они в словаре, и проверьте соответствующие пути
507	16.10	Решение 2: а если отсортировать годы? По какому критерию проводить сортировку?

508	17.9	Как выглядит решение методом «грубой силы» для нахождения $k$ -го наименьшего значения $3^a * 5^b * 7^c$ ?
509	17.12	Попробуйте применить рекурсивный подход
510	17.26	Решение 1: постарайтесь найти алгоритм вычисления степени сходства двух документов за время $O(A + B)$
511	17.24	Решение методом «грубой силы» требует постоянно вычислять суммы каждой матрицы. Как оптимизировать этот процесс?
512	17.7	Попробуйте создать отображение имен на их «правильное» написание. Также необходимо хранить информацию о соответствии «правильного» написания и всех синонимов. Иногда будет возникать необходимость в слиянии двух групп имен. Поэкспериментируйте с этим алгоритмом и посмотрите, удастся ли вам заставить его работать. После этого займитесь упрощением/оптимизацией
513	16.7	Если значение $k$ равно 1, когда $a > b$ , и 0 в противном случае, можно возвращать $a*k + b*(not\ k)$ . Но как создать $k$ ?
514	16.10	Решение 2: так ли необходимо хранить связь между годами рождения и смерти? Важно ли, в какой год умер конкретный человек, или можно обойтись простым списком годов смерти?
515	17.5	Начните с решения методом «грубой силы»
516	17.16	Рекурсивное решение: можно оптимизировать это решение при помощи мемоизации. Каким временем выполнения будет характеризоваться это решение?
517	16.3	Как найти точку пересечения двух линий? Если два отрезка пересекаются, то это должна быть та же точка, что и у их «бесконечных» продолжений. Находится ли точка пересечения внутри обоих отрезков?
518	17.26	Какая связь существует между пересечением и объединением? Можно ли вычислить одно, зная другое?
519	17.20	Вспомните, что медианой называется значение, которое больше одной половины чисел и меньше другой половины
520	16.14	Проверить все возможные линии не удастся — их количество бесконечно. Но вы знаете, что «лучшая» линия должна проходить минимум через две точки. Можно ли соединить каждую пару точек? Можно ли проверить, является ли каждая линия лучшей?
521	16.26	Можно ли обработать выражение слева направо? В каком случае это решение может не сработать?
522	17.10	Начните с решения методом «грубой силы». Можно ли просто проверить каждое значение и узнать, является ли оно доминирующим элементом?
523	16.10	Решение 2: люди в данном случае «взаимозаменяемы»: неважно, кто когда родился и когда умер. Все, что нужно, — список годов рождения и смерти. Такой подход может упростить выбор способа сортировки списка людей
524	16.25	Сначала определите масштаб задачи. Какие возможности должна предоставлять ваша реализация?
525	17.24	Возможно ли провести предварительную обработку, в результате которой сумма подматрицы будет вычисляться за время $O(1)$ ?
526	17.16	Рекурсивное решение: время выполнения решения с мемоизацией должно быть равно $O(N)$ при затратах памяти $O(N)$
527	16.3	Тщательно продумайте обработку отрезков с одинаковым углом наклона и точкой пересечения с осью $y$
528	16.13	Чтобы линия делила два квадрата надвое, она должна проходить через центры обоих квадратов

529	16.14	Постарайтесь найти решение $O(N^2)$
530	17.14	Рассмотрите возможность реорганизации или введения дополнительных структур данных
531	16.17	Представьте массив как чередующиеся последовательности положительных и отрицательных чисел. Заметьте, что искомая последовательность никогда не включает часть положительной или часть отрицательной последовательности
532	16.10	Решение 2: попробуйте создать список рождений и список смертей. Можно ли перебрать оба списка для отслеживания количества людей, живущих в любой конкретный момент времени?
533	16.22	Вариант 2: подумайте над тем, как работает ArrayList. Можно ли использовать ArrayList в данном случае?
534	17.26	Чтобы понять связь между объединением и пересечением двух множеств, рассмотрите диаграмму Венна (диаграмма с перекрывающимися кругами)
535	17.22	Получив решение методом «грубой силы», попробуйте найти более быстрый способ получения всех действительных слов, находящихся на расстоянии одного изменения. Не стоит генерировать все строки, находящиеся на расстоянии одного изменения, если подавляющее большинство таких строк не является допустимыми словами
536	16.2	Можно ли применить хеш-таблицу для оптимизации?
537	17.7	В более простой реализации описанного подхода каждое имя связывается со списком альтернативных написаний. Что произойдет, если имени в одной группе присваивается имя из другой группы?
538	17.11	Можно построить таблицу, связывающую каждое слово со списком позиций, в которых оно встречается. Как найти две ближайшие позиции?
539	17.24	А если вычислить сумму подматрицы, начинающейся с левого верхнего угла и распространяющейся до каждой ячейки? Сколько времени потребуется для ее вычисления? А если это будет сделано, можно ли вычислить сумму произвольной подматрицы за время $O(1)$ ?
540	16.22	Вариант 2: использовать ArrayList можно, но это будет весьма хлопотно. Возможно, будет проще создать собственную реализацию, специализированную для матриц
541	16.10	Решение 3: каждое рождение добавляет одного человека, а каждая смерть исключает одного человека. Попробуйте написать пример списка (с датами рождения и смерти), а затем переформатируйте его в список годов с +1 для каждого рождения и -1 для каждой смерти
542	17.16	Итеративное решение: возьмите рекурсивное решение и проанализируйте его подробнее. Можно ли реализовать аналогичную стратегию в итеративной форме?
543	17.15	Расширьте предыдущую идею для нескольких слов. Можно ли просто разбить каждое слово всеми возможными способами?
544	17.1	Двоичное сложение можно представить как перебор битов числа с суммированием битов и переносом единицы в случае необходимости. Также можно рассматривать операцию с точки зрения группировки операций. Что если начать с суммирования всех битов (без переносов), а потом обработать переносы?
545	16.21	Проведите вычисления или поэкспериментируйте с примерами. Как должна выглядеть пара? Что можно сказать о ее значениях?

546	17.20	Учтите, что все сгенерированные элементы нужно где-то хранить. Даже наименьший из первых 100 элементов может стать медианой. Отбросить очень большие или очень малые элементы не удастся
547	17.26	Решение 2: возникает искушение применить второстепенные оптимизации — например, отслеживать минимальные и максимальные элементы в каждом массиве. В некоторых случаях это позволит быстро определить, перекрываются ли массивы. Недостаток такого подхода (и других оптимизаций по тому же образцу) заключается в том, что вам все равно придется сравнивать все документы со всеми остальными документами. Факт минимализма степени сходства в нем никак не используется. Так как количество документов велико, следует избегать сравнения всех документов со всеми остальными документами (даже если такое сравнение выполняется очень быстро). Все решения такого рода имеют сложность $O(D^2)$ , где $D$ — количество документов, поэтому сравнивать все документы со всеми остальными документами нежелательно
548	16.24	Начните с решения методом «грубой силы». Каким временем выполнения оно характеризуется? Как выглядит лучшее ожидаемое время выполнения для этой задачи?
549	16.10	Решение 3: а если создать массив годов и изменений населения за каждый год? Как найти год с наибольшим количеством живущих людей?
550	17.9	При поиске $k$ -го наименьшего значения $3^a * 5^b * 7^c$ мы знаем, что $a$ , $b$ и $c$ меньше либо равны $k$ . Как сгенерировать все такие числа?
551	16.17	Заметим, что если имеется последовательность значений с отрицательной суммой, она не может находиться в начале или конце искомой последовательности. (Она может входить в искомую последовательность только в том случае, если она разделяет две другие последовательности.)
552	17.14	Нельзя ли отсортировать числа?
553	16.16	Можно представить, что массив разделен на три подмассива: LEFT, MIDDLE, RIGHT. И LEFT, и RIGHT отсортированы, а элементы MIDDLE следуют в произвольном порядке. Необходимо расширять подмассив MIDDLE до тех пор, пока сортировка входящих в него элементов не обеспечит сортировки всего массива
554	17.16	Итеративное решение: вероятно, будет проще начать с конца массива и двигаться в обратном направлении
555	17.26	Решение 2: если сравнение всех документов со всеми остальными документами нежелательно, придется перейти на уровень элементов. Рассмотрите наивное решение и посмотрите, не удастся ли расширить его для нескольких документов
556	17.22	Чтобы быстро получить все действительные слова, находящиеся на расстоянии одного изменения, попробуйте применить осмысленную группировку слов в словаре. Заметьте, что все слова по шаблону <code>b_ll</code> (такие, как <code>bill</code> , <code>ball</code> , <code>bell</code> и <code>bull</code> ) находятся на расстоянии одного изменения. Однако это не все слова, которые находятся на расстоянии одного изменения от <code>bill</code>
557	16.21	При перемещении значения $a$ из массива $A$ в массив $B$ сумма $A$ уменьшается на $a$ , а сумма $B$ увеличивается на $a$ . Что происходит, когда вы меняете местами два значения? При каком условии вы поменяете местами два значения и получите ту же сумму?
558	17.11	Если у вас имеется список вхождений каждого слова, то фактически вы ищете пару значений в двух массивах (по одному на каждый массив) с наименьшей разностью. Такой алгоритм может быть очень похож на исходный



559	16.22	Вариант 2: в одном из возможных решений размер массива удваивается, когда муравей подходит к краю. Но как решить проблему перехода в область отрицательных координат? Индексы массива не могут быть отрицательными						
560	16.13	Если имеется линия (заданная углом наклона и точкой пересечения оси $y$ ), как определить, в какой точке она пересекается с другой линией?						
561	17.26	Решение 2: фактически нужно очень быстро получить список всех документов, обладающих определенной степенью сходства с конкретным документом. (И снова не следует действовать по принципу «просмотреть все документы и быстро исключить непохожие документы» — это потребует времени не менее $O(D^2)$ .)						
562	17.16	Итеративное решение: заметьте, что три сеанса подряд ни при каких условиях пропускаться не будут. Почему? В расписание всегда можно включить средний сеанс						
563	16.14	А вы пытались воспользоваться хеш-таблицей?						
564	16.21	Если поменять местами два значения $a$ и $b$ , то сумма $A$ принимает значение, равное $\text{sum}A - a + b$ , а сумма $B$ — $\text{sum}B - b + a$ . Эти суммы должны быть равны						
565	17.24	<p>Если заранее вычислить сумму от левого верхнего угла до каждой ячейки, собранная информация может использоваться для вычисления суммы произвольной подматрицы за время <math>O(1)</math>. Представьте конкретную подматрицу. Полная заранее вычисленная сумма включает эту подматрицу, массив, находящийся над ней (<math>C</math>), массив слева (<math>B</math>) и область слева и сверху (<math>A</math>). Как вычислить сумму только для <math>D</math>?</p> <div style="text-align: center;"> <math>x1</math>            <math>x2</math> </div> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding-right: 10px;"><math>y1</math></td> <td style="border: 1px solid black; padding: 5px; text-align: center;">A</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">C</td> </tr> <tr> <td style="padding-right: 10px;"><math>y2</math></td> <td style="border: 1px solid black; padding: 5px; text-align: center;">B</td> <td style="border: 1px solid black; padding: 5px; text-align: center;">D</td> </tr> </table>	$y1$	A	C	$y2$	B	D
$y1$	A	C						
$y2$	B	D						
566	17.10	Рассмотрим решение методом «грубой силы». Мы выбираем элемент и проверяем, является ли он доминирующим, сравнивая количество совпадающих и несовпадающих элементов. Допустим, для первого элемента после нескольких первых проверок обнаруживаются 7 несовпадающих и 3 совпадающих элемента. Нужно ли продолжать проверку элемента?						
567	16.17	Начните от начала массива. С увеличением подпоследовательность остается лучшей. Но стоит подпоследовательности стать отрицательной, как она становится бесполезной						
568	17.16	Итеративное решение: если включить в расписание сеанс $i$ , то сеанс $i + 1$ включаться ни при каких условиях не будет, но всегда будет включаться сеанс $i + 2$ или $i + 3$						
569	17.26	Решение 2: развивая идею предыдущей подсказки, можно спросить, что определяет список документов с некоторой степенью сходства с документом — например, $\{13, 16, 21, 3\}$ . Какими атрибутами должен обладать такой список? Как собрать все подобные документы?						

570	16.22	Вариант 2: заметим, что в задаче не требуется, чтобы метки задач оставались неизменными. Можно ли переместить муравья и все ячейки в область положительных координат? Другими словами, что произойдет, если при необходимости разрастания массива в отрицательном направлении все индексы будут смещены так, чтобы все они остались положительными?
571	16.21	Требуется найти значения $a$ и $b$ , для которых $\text{sum}A - a + b = \text{sum}B - b + a$ . Математические преобразования помогут вам понять, что это означает для $a$ и $b$
572	16.9	Действуйте последовательно, начиная с вычитания. После того как одна функция будет закончена, вы можете использовать ее для реализации других
573	17.6	Начните с решения методом «грубой силы»
574	16.23	Начните с решения методом «грубой силы». Сколько раз в нем будет вызываться <code>rand5</code> в худшем случае?
575	17.20	Другой возможный подход к решению: можно ли отслеживать нижнюю и верхнюю половину элементов?
576	16.10	Решение 3: будьте внимательны с мелочами. Правильно ли ваш алгоритм/код обрабатывает людей, умерших в год рождения? Такие люди должны увеличивать счетчик населения на 1
577	17.26	Решение 2: список документов, сходных с $\{13, 16, 21, 3\}$ , включает все документы с элементами 13, 16, 21 и 3. Как эффективно найти этот список? Помните, что эта операция будет выполняться со многими документами, так что есть смысл провести предварительную обработку
578	17.16	Итеративное решение: используйте пример и двигайтесь в обратном направлении. Оптимальное решение для подмассивов $\{r_n\}, \{r_{n-1}, r_n\}, \{r_{n-2}, \dots, r_n\}$ находится легко. Как использовать полученную информацию для быстрого нахождения оптимального решения для $\{r_{n-3}, \dots, r_n\}$ ?
579	17.2	Допустим, у вас имеется метод <code>shuffle</code> , который работает для колод, содержащих до $n - 1$ элементов. Можно ли воспользоваться этим методом для реализации нового метода <code>shuffle</code> , который работает для колод, содержащих до $n$ элементов?
580	17.22	Создайте отображение, связывающее универсальную форму (вида <code>b_ll</code> ) со всеми словами в этой форме. В этом случае, если потребуется найти все слова, находящиеся на расстоянии одного изменения от <code>bill</code> , найдите в отображении соответствия для <code>_ill</code> , <code>b_ll</code> , <code>bi_l</code> и <code>bil_</code>
581	17.24	Сумма $D$ будет равна $\text{sum}(A\&B\&C\&D) - \text{sum}(A\&B) - \text{sum}(A\&C) + \text{sum}(A)$
582	17.17	Нельзя ли воспользоваться нагруженным деревом ( <code>trie</code> )?
583	16.21	После преобразования выясняется, что мы ищем пару таких значений, что $a - b = (\text{sum}A - \text{sum}B) / 2$ . Задача сводится к поиску пары значений с конкретной разностью
584	17.26	Решение 2: попробуйте построить хеш-таблицу, связывающую каждое слово с документами, содержащими это слово. Это позволит легко найти все документы, обладающие сходством с $\{13, 16, 21, 3\}$
585	16.5	Откуда может взяться завершающий нуль в результате $n!$ ? Что это означает?

586	17.7	Если каждое имя отображается на список альтернативных вариантов написания, назначение X и Y синонимами может потребовать обновления множества списков. Если X является синонимом {A, B, C}, а Y является синонимом {D, E, F}, то вам придется добавить {Y, D, E, F} в список синонимов A, в список синонимов B, в список синонимов C и в список синонимов X. То же самое придется проделать с {Y, D, E, F}. Нельзя ли ускорить обновление информации?
587	17.16	Итеративное решение: если сеанс включается в расписание, то следующий сеанс включаться не может, но возможно включение любого из следующих сеансов. Следующим образом, $\text{optimal}(r_i, \dots, r_n) = \max(r_i + \text{optimal}(r_{i+2}, \dots, r_n), \text{optimal}(r_{i+1}, \dots, r_n))$ . Задачу можно решить итеративно в обратном направлении
588	16.8	А вы рассматривали отрицательные числа? Подойдет ли ваше решение для таких значений, как 100 030 000?
589	17.15	Если итеративный алгоритм работает крайне неэффективно, попробуйте выделить повторяющиеся подзадачи
590	17.19	Часть 1: если нужно найти отсутствующее число с затратами времени $O(N)$ и затратами памяти $O(1)$ , значит, можно сделать ограниченное количество проходов по массиву и сохранить значения ограниченного числа переменных
591	17.9	Присмотритесь к списку всех значений $3^a * 5^b * 7^c$ . Заметьте, что каждое значение в списке будет равно $3 * (\text{некое предыдущее значение})$ , $5 * (\text{некое предыдущее значение})$ или $7 * (\text{некое предыдущее значение})$
592	16.21	Решение методом «грубой силы» просто перебирает все пары значений и ищет пару с нужной разностью. Вероятно, при этом будет использоваться внешний цикл по A с внутренним циклом по B. Для каждого значения вычисляется разность, которая сравнивается с искомым значением. Но нельзя ли действовать более конкретно? Нельзя ли для известного значения A и целевой разности определить точное значение искомого элемента B?
593	17.14	Нельзя ли воспользоваться кучей или деревом?
594	16.17	Если вы отслеживаете накапливаемую сумму, она должна сбрасываться сразу же, как только подпоследовательность станет отрицательной. Отрицательная последовательность никогда не должна включаться в начало и конец другой подпоследовательности
595	17.24	С предварительной обработкой можно обеспечить время выполнения $O(N^4)$ . Возможно ли улучшить этот результат?
596	17.3	Попробуйте применить рекурсию. Предположим, имеется алгоритм для получения подмножества размером m из n - 1 элементов. Можно ли разработать алгоритм для получения подмножества размером m из n элементов?
597	16.24	Можно ли ускорить выполнение с помощью хеш-таблицы?
598	17.22	Вероятно, ваш предыдущий алгоритм напоминал поиск в глубину. Можно ли ускорить его?
599	16.22	Вариант 3: подумайте, так ли уж нужна сетка для реализации задачи. Какая информация действительно необходима?
600	16.9	Вычитание: не поможет ли функция отрицания (преобразующая положительное число в отрицательное)? Возможно ли реализовать ее с использованием оператора сложения?
601	17.1	Сосредоточьтесь на одном из шагов описанной выше схемы. Если вы «забудете» о переносе единиц, как будет выглядеть операция сложения?

602	16.21	По сути алгоритм «грубой силы» ищет в В значение, равное $a$ - <i>целая_сумма</i> . Как быстрее найти этот элемент? Что позволяет быстрее узнать, существует ли элемент в массиве?
603	17.26	Решение 2: после того, как у вас появится способ ускорения поиска документов, сходных с конкретным документом, можно перебрать эти документы и вычислить степень сходства с ними по простому алгоритму. Можно ли ускорить обработку? А именно можно ли вычислить степень сходства непосредственно по хеш-таблице?
604	17.10	Доминирующий элемент на первый взгляд может и не казаться доминирующим. Например, доминирующий элемент может быть первым элементом массива, а затем ни разу не встречаться в следующих восьми элементах. Однако в таких случаях доминирующий элемент будет появляться позднее в массиве (возможно — <i>намного</i> позднее). Не обязательно продолжать проверку конкретного экземпляра элемента на доминирование, если он уже выглядит «маловероятным»
605	17.7	Вместо этого X, A, B и C должен ставиться в соответствие один экземпляр множества {X, A, B, C}. Y, D, E и F должен ставиться в соответствие один экземпляр {Y, D, E, F}. Когда X и Y назначаются синонимами, достаточно скопировать одно из этих множеств в другое (например, добавить {Y, D, E, F} в {X, A, B, C}). Какие еще изменения должны быть внесены в хеш-таблицу?
606	16.21	Можно воспользоваться хеш-таблицей. Также можно применить сортировку. Оба способа ускоряют поиск элементов
607	17.16	Итеративное решение: если внимательно следить за тем, какие данные действительно необходимы, вам удастся решить задачу за время $O(n)$ в $O(1)$ дополнительного пространства
608	17.12	Взгляните на это так: допустим, имеются методы <code>convertLeft</code> и <code>convertRight</code> (преобразующие левое и правое поддерево в двусвязные списки). Удастся ли использовать эти методы для преобразования всего дерева в двусвязный список?
609	17.19	Часть 1: а если просуммировать все значения в массиве? Возможно ли определить отсутствующее число?
610	17.4	За какое время можно определить младший бит отсутствующего числа?
611	17.26	Решение 2: представьте, что вы ищете документы, сходные с {1, 4, 6}, с использованием хеш-таблицы, связывающей слово с документами. Один идентификатор документа несколько раз встречается в процессе поиска. О чем это говорит?
612	17.6	Вместо того чтобы считать количество «двоек» в каждом числе, попробуйте мыслить на уровне разрядов. Другими словами, подсчитайте количество двоек в первом разряде (для каждого числа), затем количество двоек во втором разряде (для каждого числа)
613	16.9	Умножение: умножение достаточно легко реализуется на базе сложения. Но что делать с отрицательными числами?
614	16.17	Задача решается с затратами времени $O(N)$ и затратами памяти $O(1)$
615	17.24	Предположим, перед вами обычный массив. Как вычислить подмассив с наибольшей суммой? См. задачу 16.17
616	16.22	Вариант 3: все, что в действительности необходимо, — это возможность определения цвета ячейки (черная или белая) и, конечно, позиции муравья. Можно ли хранить список только белых ячеек?
617	17.17	Одно из возможных решений — вставка каждого суффикса большей строки в нагруженное дерево. Например, для слова <code>dogs</code> будут использоваться суффиксы <code>dogs</code> , <code>ogs</code> , <code>gs</code> и <code>s</code> . Как это поможет в решении задачи? Как будет выглядеть время выполнения?

618	17.22	Поиск в ширину часто выполняется быстрее поиска в глубину — не обязательно в худшем случае, но во многих других случаях. Почему? Можно ли найти еще более быстрое решение?
619	17.5	А если начать с начала, подсчитывая вхождения $A$ и $B$ , встреченные до настоящего момента? (Попробуйте построить таблицу с индексами массива и количествами обнаруженных $A$ и $B$ .)
620	17.10	Доминирующий элемент должен быть доминирующим элементом для некоторого подмассива, и никакой подмассив не может иметь нескольких доминирующих элементов
621	17.24	Предположим, нужно просто найти максимальную подматрицу, начинающуюся в строке $r_1$ и завершающуюся в строке $r_2$ . Как сделать это наиболее эффективно? (См. предыдущую подсказку.) А если теперь потребуется найти максимальный подмассив от $r_1$ до $(r_2+2)$ , как это сделать эффективно?
622	17.9	Поскольку каждое число равно предыдущему значению в списке, умноженному на 3, 5 или 7, можно просто проверить все возможные значения и выбрать следующее из еще не встречавшихся. Это приведет к значительному дублированию работы. Как этого избежать?
623	17.13	Можно ли опробовать все возможности? Как это может выглядеть?
624	16.26	Умножение и деление относятся к высокоприоритетным операциям. В выражении вида $3*4 + 5*9/2 + 3$ операции умножения и деления должны быть сгруппированы
625	17.14	Если выбрать произвольный элемент, сколько времени потребуется для определения ранга этого элемента (количества элементов, больших или меньших его)?
626	17.19	Часть 2: теперь мы ищем два отсутствующих числа, которые будут обозначены $a$ и $b$ . Решение из части 1 позволит узнать сумму $a$ и $b$ , но не сами значения $a$ и $b$ . Какие еще вычисления можно проделать?
627	16.22	Вариант 3: можно рассмотреть возможность хранения хешированного множества всех белых ячеек. Но как тогда вывести всю сетку?
628	17.1	Суммирование само по себе выполняет преобразования: $1 + 1 \rightarrow 0, 1 + 0 \rightarrow 1, 0 + 1 \rightarrow 1, 0 + 0 \rightarrow 0$ . Как сделать то же без знака $+$ ?
629	17.21	Какую роль играет самый высокий столбец в гистограмме?
630	16.25	Какая структура данных лучше всего подойдет для выборки? Какая структура данных наиболее эффективна для определения и поддержания порядка элементов?
631	16.18	Начните с решения методом «грубой силы». Можно ли опробовать все возможные варианты для $a$ и $b$ ?
632	16.6	А если отсортировать массивы?
633	17.11	Нельзя ли обойти массивы с использованием двух указателей? Обход должен быть выполнен за время $O(A+B)$ , где $A$ и $B$ — размеры двух массивов
634	17.2	Алгоритм можно построить рекурсивно, меняя местами $n$ -й элемент с любым из предшествующих элементов. Как эта процедура будет выглядеть в итеративном виде?
635	16.21	Допустим, сумма $A$ равна 11, а сумма $B$ равна 8. Может ли существовать пара с нужной разностью? Убедитесь в том, что ваше решение правильно обрабатывает подобные ситуации

636	17.26	Существует альтернативное решение: вы берете все слова из всех документов, помещаете их в один огромный список и сортируете этот список. Предполагается, что после этого вы знаете, к какому документу относится каждое слово. Как отслеживать сходные пары?
637	16.23	<p>Постройте таблицу, показывающую, как возможные варианты вызовов <code>rand5()</code> будут отображаться на результат <code>rand7()</code>. Например, если реализовать <code>rand3()</code> по формуле <math>(\text{rand2}() + \text{rand2}()) \% 3</math>, то таблица будет выглядеть так, как показано ниже. Проанализируйте таблицу. Какие выводы можно сделать?</p> <p><b>Первый вызов</b>  <b>Второй вызов</b>  <b>Результат</b></p>
		<pre> 0 0 0  0 1 1  1 0 1  1 1 2 </pre>
638	17.8	В этой задаче требуется найти самую длинную последовательность пар, в которой обе стороны пары непрерывно возрастают. А если возрастать должна только одна сторона пары?
639	16.15	Для начала постройте массив с частотами вхождений всех элементов
640	17.21	Выберите самый высокий столбец, затем самые высокие столбцы слева и справа от него. Вода заполнит эти области. Можно ли вычислить ее объем? Что делать с остальными областями?
641	17.6	Существует ли более быстрый способ вычисления количества двоек в определенном разряде по диапазону чисел? Двойки должны составлять примерно 1/10 всех цифр, но только примерно. Как уточнить эту оценку?
642	17.1	Для реализации сложения можно воспользоваться операцией XOR
643	16.18	Заметим, что одна из подстрок, $a$ или $b$ , должна начинаться в начале строки. Это обстоятельство сокращает количество возможностей
644	16.24	А если массив отсортирован?
645	17.18	Начните с решения методом «грубой силы»
646	17.12	Когда вы начнете в общих чертах представлять рекурсивный алгоритм, возникает следующая проблема: иногда рекурсивный алгоритм должен возвращаться к началу связанного списка, а иногда он должен возвращаться к концу. У этой проблемы есть несколько решений. Попробуйте найти некоторые из них
647	17.14	Если выбрать произвольный элемент, в среднем этот элемент будет находиться где-то около 50-й перцентили (половина элементов выше, половина элементов ниже). А если это будет происходить многократно?

648	16.9	Деление: если вы пытаетесь вычислить $x = a/b$ , помните, что $a = bx$ . Можно ли найти ближайшее значение для $x$ ? Помните, что речь идет о целочисленном делении, и значение $x$ должно быть целым
649	17.19	Часть 2: существует много разных вычислений, которые можно опробовать. Например, можно перемножить все числа, но тогда мы получим только произведение $a$ и $b$
650	17.10	Попробуйте действовать так: для заданного элемента начните проверять, является ли он началом подмассива, для которого он является доминирующим. Когда это становится «маловероятным» (элемент встречается менее чем в половине случаев), переходите к проверке следующего элемента (элемент после подмассива)
651	17.21	Чтобы вычислить величину области между самым высоким столбцом гистограммы и самым высоким столбцом слева от него, достаточно пройти по гистограмме и исключить все промежуточные столбцы. То же самое можно сделать справа. Как поступить с оставшейся частью гистограммы?
652	17.18	Одно из решений методом «грубой силы» для каждой начальной позиции перебирает элементы, пока не найдет подпоследовательность, содержащую все целевые символы
653	16.18	Не забудьте обработать ситуацию, в которой первым символом <code>pattern</code> является <code>b</code>
654	16.20	Мы знаем, что в реальном мире некоторые префиксы/подстроки невозможны. Например, возьмем число 33835676368. Хотя 3383 соответствует строке <code>fftf</code> , не существует слов, начинающихся с <code>fftf</code> . Возможно ли обойти такие ситуации?
655	17.7	Также можно рассматривать данные задачи как граф. Как будет работать такое решение?
656	17.13	Решения, принимаемые рекурсивным алгоритмом, можно рассматривать одним из двух способов: 1) для каждого символа: следует ли поместить здесь пробел? 2) где поместить следующий пробел? Обе разновидности могут быть решены с применением рекурсии
657	17.8	Если возрастать должна только одна сторона пары, то значения можно просто отсортировать с этой стороны. Самая длинная последовательность будет состоять из всех пар (кроме дубликатов, так как самая длинная последовательность должна строго возрастать). Что это означает для исходной задачи?
658	17.21	Оставшуюся часть гистограммы можно обработать по той же схеме: найти самый высокий столбец, найти второй по высоте столбец и исключить столбцы между ними
659	17.4	Чтобы найти младший бит отсутствующего числа, стоит заметить, что ожидаемое количество 0 и 1 известно. Например, если в младшем бите встречаются три 0 и три 1, то младший бит отсутствующего числа должен быть равен 1. Подумайте: в любой последовательности младший разряд сначала содержит 0, потом 1, потом 0, потом 1 и т. д.
660	17.9	Вместо того чтобы проверять все значения в списке (умножая каждое на 3, 5 и 7), взгляните на ситуацию под другим углом: при вставке значения $x$ в список можно «сгенерировать» значения $3x$ , $5x$ и $7x$ для использования в будущем
661	17.14	Поразмыслите над предыдущей подсказкой, особенно в контексте быстрой сортировки
662	17.21	Как ускорить процесс поиска следующего по высоте столбца с каждой из сторон?

663	16.18	Будьте внимательны с анализом времени выполнения. Если при переборе $O(n^2)$ подстрок для каждой строки выполняются $O(n)$ сравнений, то общее время выполнения составит $O(n^3)$
664	17.1	Теперь сосредоточьтесь на переносе. В каких случаях будет происходить перенос? Как применить его к числу?
665	16.26	По сути речь идет о том, что при достижении знака умножения или деления происходит переход к отдельному «процессу» для вычисления результата фрагмента
666	17.8	Если отсортировать значения по высоте, вы получите порядок итоговых пар. Самая длинная последовательность должна следовать в этом относительном порядке (но не обязательно содержать все пары). Теперь достаточно найти самую длинную возрастающую подпоследовательность по весу с сохранением относительного порядка элементов. Фактически эта задача эквивалентна поиску самой длинной последовательности в массиве целых чисел (без изменения порядка элементов)
667	16.16	Рассмотрим три подмассива: LEFT, MIDDLE, RIGHT. Пока ограничимся одним вопросом: возможно ли отсортировать MIDDLE так, что весь массив будет отсортирован? Как это проверить?
668	16.23	Снова взглянув на таблицу, мы видим, что количество строк равно $5^k$ , где $k$ — максимальное количество вызовов <code>rand5()</code> . Чтобы все значения в диапазоне от 0 до 6 имели равную вероятность, $1/7$ строк должна соответствовать 0, еще $1/7$ — 1, и т. д. Возможно ли это?
669	17.18	На решение методом «грубой силы» можно взглянуть иначе: мы берем каждый начальный индекс и ищем следующий экземпляр каждого элемента в целевой строке. Максимальное из следующих вхождений отмечает конец подпоследовательности, содержащей все целевые символы. Каким временем выполнения характеризуется это решение? Как ускорить его работу?
670	16.6	Подумайте над тем, как бы вы провели слияние двух отсортированных массивов
671	17.5	Если в приведенных таблицах содержатся равные значения A и B, весь подмассив (начиная с индекса 0) содержит равные количества A и B. Как использовать эту таблицу для поиска подмассивов, не начинающихся с индекса 0?
672	17.19	Часть 2: сложение чисел позволит получить результат $a + b$ . Умножение позволит получить результат $a * b$ . Как получить точные значения $a$ и $b$ ?
673	16.24	Если отсортировать массив, мы могли бы найти дополнение числа посредством бинарного поиска. А если бы массив уже был получен отсортированным? Возможно ли было решить задачу с затратами времени $O(N)$ и затратами памяти $O(1)$ ?
674	16.19	Если вам известны строка и столбец ячейки, содержащей воду, как найти все соседние области?
675	17.7	Добавление X, Y эквивалентно добавлению ребра между узлом X и узлом Y. Как найти группы синонимов?
676	17.21	Можно ли выполнить предварительную обработку для вычисления следующего по высоте столбца с каждой стороны?
677	17.13	Будет ли рекурсивный алгоритм многократно повторять одни подзадачи? Можно ли оптимизировать его с помощью хеш-таблицы?



678	17.14	А если при выборе элемента другие элементы будут меняться местами (как при быстрой сортировке), так чтобы элементы, находящиеся под ним, предшествовали элементам, находящимся выше? Если проделать это многократно, сможете ли вы найти миллион наименьших чисел?
679	16.6	Допустим, имеются два отсортированных массива и вы перебираете их элементы. Если указатель в первом массиве установлен на 3, а указатель во втором массиве установлен на 9, то как перемещение второго указателя отразится на разности пары?
680	17.12	Чтобы рекурсивный алгоритм мог выбирать между началом и концом связанного списка, можно попытаться передать параметр-флаг. Впрочем, такое решение работает не очень хорошо. Проблема в том, что при вызове <code>convert(current.left)</code> требуется получить конец связанного списка <code>left</code> ; это позволит присоединить конец связанного списка к <code>current</code> . Но если <code>current</code> является чьим-то правым поддеревом, вызов <code>convert(current)</code> должен передать начало связанного списка (которое в действительности является началом связанного списка <code>current.left</code> ). На самом деле нужно знать как начало, так и конец связанного списка
681	17.18	Вернемся к рассмотренному ранее решению методом «грубой силы». Его узким местом является многократный запрос следующего вхождения конкретного символа. Можно ли оптимизировать эту операцию? Это нужно сделать за время $O(1)$
682	17.8	Попробуйте применить рекурсивный подход, который просто проверяет все возможности
683	17.4	После того как вы определили, что младший бит равен 0 (или 1), можно исключить все числа, младший бит которых равен 0. Чем эта задача отличается от предыдущей части?
684	17.23	Начните с решения методом «грубой силы». Можно ли сначала проверить наибольший возможный квадрат?
685	16.18	Предположим, вы выбрали конкретное значение для части <code>a</code> шаблона. Сколько возможностей существует для <code>b</code> ?
686	17.9	При добавлении <code>x</code> в список первых <code>k</code> значений можно добавить <code>3x</code> , <code>5x</code> и <code>7x</code> в новый список. Как проделать это с максимальной эффективностью? Стоит ли поддерживать несколько очередей со значениями? Всегда ли нужно вставлять <code>3x</code> , <code>5x</code> и <code>7x</code> ? А может, в каких-то случаях достаточно вставить только <code>7x</code> ? Одно число не должно встречаться дважды
687	16.19	Попробуйте применить рекурсию для подсчета количества ячеек с водой
688	16.8	Попробуйте разделить число на группы из трех цифр
689	17.19	Часть 2: можно выполнить обе операции. Если мы знаем, что $a + b = 87$ , $a * b = 962$ , эти уравнения можно решить для <code>a</code> и <code>b</code> : $a = 13$ и $b = 74$ . Однако при этом придется перемножать достаточно большие числа. Произведение всех чисел может быть больше $10^{157}$ . Нельзя ли найти более простые вычисления?
690	16.11	Задумайтесь над процессом построения трамплина. Какие решения вам придется принимать?
691	17.18	Можно ли заранее найти следующий экземпляр символа для каждого индекса? Попробуйте воспользоваться многомерным массивом
692	17.1	Перенос происходит при выполнении сложения $1 + 1$ . Как применить перенос к числу?

693	17.21	Альтернативное решение: взгляните на ситуацию с точки зрения каждого столбца. Каждый столбец будет накрыт водой. Сколько воды будет покрывать каждый столбец?
694	16.25	Пригодится и хеш-таблица, и двусвязный список. Можно ли объединить их?
695	17.23	Наибольший возможный квадрат имеет размеры $N \times N$ . Итак, если проверить этот квадрат и он подойдет, вы знаете, что был найден лучший квадрат. В противном случае можно переходить к проверке следующего по размерам квадрата
696	17.19	Часть 2: подойдет почти любая формула (не являющаяся эквивалентом линейной суммы). Вопрос в том, чтобы результат оставался небольшим
697	16.23	Разделить $5^k$ на 7 без остатка невозможно. Означает ли это, что <code>rand7()</code> невозможно реализовать на базе <code>rand5()</code> ?
698	16.26	Также можно вести два стека: для операторов и для чисел. Каждый раз, когда вы встречаете число, оно заносится в стек. Как насчет операторов? Когда операторы должны извлекаться из стека и применяться к числам?
699	17.8	Если имеется самая длинная последовательность, заканчивающаяся на каждом элементе от $A[0]$ до $A[n-1]$ , можно ли использовать ее для поиска самой длинной последовательности, заканчивающейся на элементе $A[n-1]$ ?
700	16.11	Рассмотрите рекурсивное решение
701	17.12	Многие кандидаты останавливаются на этом и не знают, что делать дальше. Иногда нужно начало связного списка, иногда нужен конец. Конкретный узел не всегда знает, что возвращать при вызове <code>convert</code> . Иногда самым лучшим оказывается самое простое решение: всегда возвращать и то и другое. Как это можно сделать?
702	17.19	Часть 2: проверьте сумму квадратов значений
703	16.20	Нагруженное дерево поможет ускорить обработку. А если в нагруженном дереве будет храниться весь список слов?
704	17.7	Каждый связный подграф представляет группу синонимов. Для поиска каждой группы можно воспользоваться многократным поиском в ширину (или в глубину)
705	17.23	Оцените время выполнения решения методом «грубой силы»
706	16.19	Как убедиться в том, что одни и те же ячейки не посещаются многократно? Подумайте, как работает поиск в глубину или в ширину в графе
707	16.7	Если $a > b$ , то $a - b > 0$ . Как получить знаковый бит $a - b$ ?
708	16.16	Чтобы в результате сортировки <code>MIDDLE</code> весь массив оказался отсортированным, должны выполняться условия $\text{MAX}(\text{LEFT}) \leq \text{MIN}(\text{MIDDLE}, \text{RIGHT})$ и $\text{MAX}(\text{LEFT}, \text{MIDDLE}) \leq \text{MIN}(\text{RIGHT})$
709	17.20	А если использовать кучу? Или две кучи?
710	16.4	Как изменится решение, если <code>hasWon</code> будет вызываться несколько раз?
711	16.5	Каждый нуль в $n!$ соответствует делимости $n$ на 10. Что это означает?
712	17.1	Используйте операцию <code>AND</code> для вычисления переноса
713	17.5	Допустим, в этой таблице для индекса $i$ $\text{count}(A, 0 \rightarrow i) = 3$ и $\text{count}(B, 0 \rightarrow i) = 7$ . Это означает, что количество $B$ на 4 превышает количество $A$ . Если вы найдете более позднюю позицию $j$ с той же разностью ( $\text{count}(B, 0 \rightarrow j) - \text{count}(A, 0 \rightarrow j)$ ), это признак подмассива с равным количеством $A$ и $B$
714	17.23	Можно ли оптимизировать решение посредством предварительной обработки?

715	16.11	Когда вы определите рекурсивный алгоритм, подумайте над временем выполнения. Можно ли ускорить его? Как?
716	16.1	Пусть <code>diff</code> — разность <code>a</code> и <code>b</code> . Можно ли как-то использовать <code>diff</code> ? Как избавиться от временной переменной?
717	17.19	Часть 2: возможно, вам понадобится формула квадратного уравнения. Если вы ее не помните, не страшно, так как мало кто ее помнит
718	16.18	Так как значение <code>a</code> определяет значение <code>b</code> (и наоборот), и либо <code>a</code> , либо <code>b</code> должно стоять в начале <code>value</code> , существуют $O(n)$ способов разбивки
719	17.12	Вернуть начало связанного списка вместе с его концом можно несколькими способами. Можно вернуть массив из двух элементов. Можно определить новую структуру данных для хранения начала и конца. Можно воспользоваться структурой данных <code>BiNode</code> . Некоторые языки (например, Python) позволяют вернуть несколько значений. Наконец, проблему можно решить при помощи циклического связанного списка. Проанализируйте эти решения. Какое из них вам нравится больше и почему?
720	16.23	Реализовать <code>rand7()</code> на базе <code>rand5()</code> возможно, просто это не сделать детерминированно (то есть с гарантией того, что вычисления завершатся после определенного количества вызовов). С учетом этого обстоятельства напишите решение, которое будет работать
721	17.23	Сделайте это за время $O(N^3)$ , где $N$ — длина одной стороны квадрата
722	16.11	Примените мемоизацию для оптимизации времени выполнения. Тщательно подумайте, что именно должно кэшироваться. Как оценить время выполнения? Оно тесно связано с максимальным размером таблицы
723	16.19	У вас должен получиться алгоритм с временем выполнения $O(N^2)$ для таблицы $N \times N$ . Если не получился, значит, вы либо неверно оценили время выполнения, либо ваш алгоритм не оптимален
724	17.1	Возможно, операцию суммирования/переноса придется выполнять многократно. Прибавление переноса к сумме может потребовать переноса новых значений
725	17.18	Когда вы сформулируете решение с предварительной обработкой, подумайте, как сократить затраты памяти. Постарайтесь добиться затрат времени $O(SB)$ и затрат памяти $O(B)$ (где $B$ — размер большего массива, а $S$ — размер меньшего массива)
726	16.20	Вероятно, алгоритм будет выполняться многократно. Можно ли оптимизировать его за счет увеличения объема предварительной обработки?
727	16.18	Алгоритм должен иметь сложность $O(n^2)$
728	16.7	А вы рассмотрели возможность целочисленного переполнения в <code>a - b</code> ?
729	16.5	Каждый множитель 10 в $n!$ означает, что $n!$ делится на 5 и на 2
730	16.15	Для простоты и четкости реализации стоит использовать другие методы и классы
731	17.18	Взгляните на ситуацию иначе: предположим, у вас есть список индексов, под которыми появлялся каждый элемент. Можно ли найти первую возможную подпоследовательность, содержащую все элементы? А вторую?
732	16.4	А если алгоритм разрабатывается для поля $N \times N$ , как изменится решение?
733	16.5	Можно ли подсчитать количество множителей 5 и 2? Нужно ли подсчитывать и то и другое?

734	17.21	Каждый столбец будет покрыт слоем воды, высота которого определяется минимумом среди высот самого высокого столбца слева и самого высокого столбца справа. Другими словами, $water\_on\_top[i] = \min(tallest\_bar(0 \rightarrow i), tallest\_bar(i, n))$
735	16.16	Можно ли расширять среднюю часть, пока не будет выполнено условие?
736	17.23	Можно ли заранее вычислить количество черных пикселей выше и левее заданной ячейки?
737	16.1	Также можно попытаться использовать XOR
738	17.22	А если выполнить поиск в ширину как от начального, так и от конечного слова?
739	17.13	В реальности мы знаем, что некоторые пути не ведут к словам. Например, не существует слов, начинающихся с подстроки <code>hellothisism</code> . Можно ли заранее прервать поиск на пути, который заведомо не приведет к успеху?
740	16.11	Существует альтернативное, умное и очень быстрое решение. Оно позволяет решить задачу с линейным временем без рекурсии. Как?
741	17.18	Используйте кучу
742	17.21	Задача должна решаться с затратами времени $O(N)$ и затратами памяти $O(N)$
743	17.17	Также можно вставить в нагруженное дерево каждую из меньших строк. Как это поможет в решении задачи? Как будет выглядеть время выполнения?
744	16.20	С предварительной обработкой время выборки сокращается до $O(1)$
745	16.5	А вы учли, что 25 состоит из двух множителей 5 и 5?
746	16.16	Задача должна решаться за время $O(N)$
747	16.11	Вы выбираете $K$ досок двух разных типов. Все варианты с 10 планками первого типа и 4 планками второго типа имеют одинаковую сумму. Возможно ли перебрать все существующие варианты?
748	17.25	Можно ли использовать нагруженное дерево для преждевременного прерывания недействительных путей?
749	17.13	Для преждевременного прерывания попробуйте применить нагруженное дерево