# Разбор задачи «Стикеры»

| | |
|---|---|
| Автор задачи: | Михаил Иванов |
| Подготовка тестов и решений: | Иван Сафонов |
| Автор разбора: | Иван Сафонов, Михаил Иванов |

Заметим, что если стикерами размера $s$ можно покрыть доску, то тем же количеством стикеров любого размера, большего $s$, тоже можно покрыть доску.

Это означает, что мы можем решить задачу бинарным поиском по ответу. Заведём переменные $\ell = 0$ и $r = \max(a, b)$ ($a$ — горизонтальная сторона, $b$ — вертикальная). Заметим, что стикерами размера $\ell$ нельзя покрыть доску, каким бы ни было положительное $k$, а стикерами размера $r$ можно покрыть доску (поскольку даже одного стикера точно хватит). Будем сохранять это свойство переменных $\ell$ и $r$: в первой из них будет находиться число, строго меньшее искомого ответа, а во второй — большее или равное ответа. Пока $r - \ell > 1$, будем повторять такую операцию: рассмотрим $h = \lfloor \frac{\ell+r}{2} \rfloor$. Проверим, можно ли покрыть доску стикерами размера $h$ за $k$ дней.

Минимальное количество дней, которое нужно, чтобы покрыть доску, равно $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil$. Строгое обоснование этого факта таково. Понятно, что таким количеством стикеров покрыть доску можно: для этого надо, начиная с левой стороны доски, проводить справа вертикальные прямые с отступом $s$, и, начиная с нижней стороны доски, проводить сверху горизонтальные прямые с отступом $s$. Тогда доска разобьётся на $\lceil \frac{a}{h} \rceil$ вертикальных полосок ширины $s$ (кроме самой правой — её ширина может быть не равна $s$, а меньше $s$), и она же — на $\lceil \frac{b}{h} \rceil$ горизонтальных полосок ширины $s$ (самая верхняя полоска может быть ширины меньше $s$). Попарно пересекаясь друг с другом, вертикальные и горизонтальные полоски разбивают доску на $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil$ прямоугольников со сторонами, не превышающими $s$. Каждый из них по отдельности можно покрыть квадратом $s \times s$, поэтому всю доску можно покрыть $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil$ стикерами. Чтобы доказать, что меньшего количества не хватит, разобьём доску на клеточки $1 \times 1$, пронумеруем столбцы от 0 до $a-1$ и строки от 0 до $b-1$. Теперь заштрихуем клетки, у которых обе координаты делятся на $s$. Будет заштриховано ровно $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil$ клеток, и площадь заштрихованной части тоже равна $s$. Заметим, что стикеры должны полностью покрывать эту площадь, однако каждый стикер в отдельности может покрыть не больше единицы от этой площади (площадь части стикера, лежащая в клетках, обе координаты которых делятся на $s$, равна единице, поэтому часть этой площади, которая лежит внутри доски и, соответственно, заштрихована, не больше одного), поэтому, чтобы покрыть всю эту площадь, потребуется хотя бы $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil$ стикеров.

Поэтому, если $\lceil \frac{a}{h} \rceil \lceil \frac{b}{h} \rceil < k$, присвоим $\ell = h$, а иначе присвоим $r = h$. Таким образом, по-прежнему стикера размера $\ell$ не хватит, чтобы покрыть доску, а стикера размера $r$ хватит. При этом величина $r - \ell - 1$ уменьшится хотя бы в два раза и останется целой. Изначально она была не больше $10^{18}$, а за 60 операций она уменьшится хотя бы в $2^{60} > 10^{18}$ раз и останется целой — следовательно, она будет равна нулю. Таким образом, в конце этого процесса мы получим, что $r = \ell + 1$, то есть $r$ будет равно минимальному размеру стикеров, которыми можно за $k$ дней покрыть всю доску, и $r$ надо будет вывести в качестве ответа.

Такое решение работает за 60 операций (а на самом деле за $\log_2 \max(a, b)$, если остановиться сразу, когда $r - \ell$ окажется равным одному). Каждая операция состоит из нескольких несложных арифметических действий, поэтому решение очень быстрое.

# Разбор задачи «Большие экраны»

| | |
|---|---|
| Автор задачи: | Арсений Кириллов |
| Подготовка тестов и решений: | Арсений Кириллов |
| Автор разбора: | Арсений Кириллов |

В этой задачи требовалось сделать ровно то, что написано в условии. Сначала вывести первую строчку всех цифр, потом вторую и так далее. Что бы это было проще сделать, можно заметить, что есть всего 4 типа различных строчек, которые бывают во всех цифрах. А именно строчки «`X....`», «`....X`», «`X...X`» и «`XXXXX`». Поэтому для каждой из цифр можно было сделать массив из

7 чисел, определяющие, какая из этих строчек первая строчка цифры, вторая и так далее. И тогда для каждой из 7 строчек вывода можно просто выводить одну из возможных строчек для каждой цифры числа.

# Разбор задачи «Тест на терпение»

| | |
|---|---|
| Автор задачи: | Владислав Мильшин |
| Подготовка тестов и решений: | Владислав Мильшин |
| Автор разбора: | Владислав Мильшин |

Давайте посмотрим на бит $i$. Можно выделить 3 случая:

- Если бит $i$ в числе $a$ больше, чем в числе $b$ (то есть в $a$ стоит `1` на месте $i$, а в $b$ стоит `0` на месте $i$), тогда не существует таких целых чисел $x$ и $y$, что их побитовое И равно $a$, а побитовое ИЛИ равно $b$, потому что из условия, что бит $i$ в $a$ равен `1`, следует, что в $x$ и $y$ на месте $i$ должны стоять `1`, а из условия, что бит $i$ в $b$ равен `0` следует обратное, а именно в разряде $i$ у двух чисел должны быть `0`. В таком случае ответ $-1$.
- Если в числе $a$ и в числе $b$ он одинаковый, то в мы можем точно определить бит $i$ в числах $x$ и $y$: он равен биту в числах $a$ и $b$.
- Если в числе $a$ стоит меньшая цифра, чем в $b$, то это значит, что в одном числе из $x$ и $y$ должен стоять `1`, а в другом — `0`.

Если хоть раз выполнился первый случай, то ответа нет; во всех битах, где выполняется второй случай, двоичные цифры восстанавливаются однозначно (при этом они одинаковые, поэтому разность соответствующих разрядных слагаемых нулевая). Теперь осталось с помощью битов третьего типа набрать определённую разность или понять, что это невозможно. Мы должны в каждом таком бите поставить единицу в одно число, а ноль — в другое. Пусть $i_1 < i_2 < \cdots < i_m$ — все номера битов из третьего случая. Так как $c \geqslant 0$, в $x$ в бите $i_m$ должна стоять `1`, а в числе $y$ должен стоять `0`. Это верно, ведь, если мы поставим единицу в $y$, то разность первого и второго числа будет не больше

$$2^{i_1} + 2^{i_2} + \ldots + 2^{i_{m-1}} - 2^{i_m} \leqslant 1 + 2 + \ldots + 2^{i_m - 1} - 2^{i_m} = -1 < c.$$

Так что поставим в число $x$ `1` и в число $y$ — `0`. Нам останется набрать разность $c' = c - 2^{i_m}$. Если $c' \geqslant 0$, поступим абсолютно аналогично, но уже с набором битов $i_1 < i_2 < \ldots < i_{m-1}$; если же $c' < 0$, то заменим $c'$ на $-c'$ и поменяем местами $x$ и $y$. Будем так продолжать действовать, пока все биты третьего типа не закончатся, то есть пока не окажется $m = 0$. Если оказалось, что полученная к этому моменту $c$ ненулевая, то мы не достигли нашей цели, поэтому это было вообще невозможно сделать. Если $c = 0$, то полученные $x$ и $y$ — это и есть ответ.

# Разбор задачи «Обмен»

| | |
|---|---|
| Автор задачи: | Макар Селиванов |
| Подготовка тестов и решений: | Макар Селиванов |
| Автор разбора: | Макар Селиванов |

Пусть $b$ — сколько следует выдать тугриков, а $a_1, a_2, \ldots, a_n$ — номиналы банкнот. Задача решается жадным алгоритмом, который заключается в следующем. Мы выдаем самую большую банкноту, которая не больше $b$, и уменьшаем $b$ на номинал этой банкноты. Эту процедуру мы повторяем, пока $b$ не станет меньше $a_1$. Если $b$ не равно нулю, то тогда нельзя выдать $b$ тугриков текущими банкотами, а иначе можно, причём мы только что сделали это минимальным числом банкнот. Докажем это по индукции по числу номиналов банкнот.

**База.** При $n = 1$ понятно, что мы не можем выдать сумму $b$ тогда и только тогда, когда $b$ не делится на $a_1$, а иначе выдаем $\frac{b}{a_1}$ банкнот.

**Индукционный переход.** Пусть $n > 1$, и для $(n-1)$ номинала банкнот мы уже доказали, что жадный алгоритм верен. Посмотрим на банкноту $a_n$. Докажем, что в оптимальном ответе будет выдано ровно $\left\lfloor \frac{b}{a_n} \right\rfloor$ банкнот этого номинала. Понятно, что их количество не может быть больше, потому что иначе Штирлиц выдаст больше, чем $b$ тугриков. Пусть в оптимальном ответе надо выдать $t$ банкнот $n$-го номинала, где $t < \left\lfloor \frac{b}{a_n} \right\rfloor$. Тогда мы должны выдать $S \geqslant a_n \cdot \left( \left\lfloor \frac{b}{a_n} \right\rfloor - t \right) \geqslant a_n$ тугриков с помощью $(n-1)$ номинала $a_1, a_2, \ldots, a_{n-1}$. Для этого количества номиналов мы уже доказали жадный алгоритм, поэтому количество выданных банкнот $(n-1)$-го номинала будет не меньше, чем $\left\lfloor \frac{S}{a_{n-1}} \right\rfloor \geqslant \left\lfloor \frac{a_n}{a_{n-1}} \right\rfloor = \frac{a_n}{a_{n-1}} > 1$. Но тогда можно все эти $\frac{a_n}{a_{n-1}}$ банкнот заменить на одну номинала $a_n$ и получить более оптимальный ответ, противоречие.

# Разбор задачи «Картошка»

| | |
|---|---|
| Автор задачи: | Михаил Иванов |
| Подготовка тестов и решений: | Михаил Иванов |
| Автор разбора: | Михаил Иванов |

Задача сделана по мотивам известной более простой задачи. Есть миска с $h$ яблоками и двое игроков, которые ходят по очереди. За ход надо взять любое целое количество яблок от 1 до $r$, проигрывает не имеющий хода. Обратите внимание, что $r$ одинаково для обоих игроков.

Давайте исследуем эту задачу. Если в миске 0 яблок, то первый игрок не может походить и проигрывает. Если в миске от одного до $r$ яблок, то первый игрок может взять их все, так что второй проиграет. Если $r+1$ яблоко, то, как бы ни походил первый, останется от 1 до $r$ яблок, и второй их все возьмёт и выиграет. Если от $r+2$ до $2r+1$, то первый игрок может взять ровно столько яблок, чтобы в миске осталось $r+1$ яблоко, и, как мы только что выяснили, второй игрок никак не выиграет, если ему досталась миска с $r+1$ яблоком и первый играет оптимально. Если в миске $2r+2$ яблока, то первый игрок своим ходом точно оставит в ней от $r+2$ до $2r+1$ яблока, то есть даст второму игроку выигрышную позицию.

Продолжая этот процесс, мы без труда поймём, что первый игрок выиграет, если $h$ не делится на $r+1$, а второй — если делится.

В чём же отличие нашей игры от этой? Во-первых, там речь идёт не о яблоках и миске, а о криках и шумовом барьере. Давайте представим, что шумовой барьер $h$ — это миска с $h$ яблоками, и Джон и Пол имеют одинаковую наибольшую силу крика $r$. Тогда криком силы $s$ игрок вынимает из миски $s$ яблок, и проигрывает тот, кто оставит в миске ни одного яблока. (Ведь в задаче заканчивалась игра, не когда игроки накричали строго сильнее шумового барьера, а когда накричали хотя бы на шумовой барьер; требовалось выполнить не неравенство $a_1 + a_2 + \ldots + a_{n-1} + a_n > h$, а неравенство $a_1 + a_2 + \ldots + a_{n-1} + a_n \geqslant h$.) То есть отличие в том, что в задаче из первого абзаца игрок проигрывает, если не может оставить количество неотрицательным, а у нас же проигрывает, если не может оставить количество строго положительным. Давайте решим эту проблему — в задаче из первого абзаца добавим в миску особое синее яблоко, которое запрещено вынимать из миски, и по-прежнему будем говорить, что проигрывает не имеющий хода (от этого задача не поменяется, ведь это синее яблоко никак не влияет на игровой процесс). Но заметим, что получится ровно вторая задача! Действительно, если игрок хочет достать сколько-то яблок из миски, но не все, то потребуем у него, чтобы он не трогал синее яблоко; если он хочет достать все яблоки, то он проиграл, так как в миске после каждого хода должно оставаться положительное число яблок.

Таким образом, вторая задача с $h$ яблоками полностью равносильна первой задаче с $h-1$ яблоком. В первой задаче с $h-1$ яблоком первый игрок выигрывал, если $h-1$ не кратно $r+1$, иначе побеждал второй. Значит, во второй задаче с $h$ яблоками первый побеждает, если $h$ имеет не остаток 1 при делении на $r+1$, а иначе второй игрок. Но это, если максимальные силы криков одинаковые. А что, если они различны: у ходящего первым Джона она $R_J$, а у ходящего вторым Пола — $R_P$?

Докажем, что почти всегда при различных максимальных силах криков выигрывает тот, у кого она больше. Во-первых, если $h = 1$, то точно выиграет второй игрок, так как у первого есть только одно синее яблоко в миске. Если $h$ от 2 до $R_J + 1$, то первый может оставить ровно одно яблоко и победить. В остальных случаях выигрывает тот, у кого строго больше $R$. Действительно, если

$R_J > R_P$, то Джон может своим ходом сделать у $h$ любой остаток при делении на $R_P + 1$ (ведь числа $h-1, h-2, \ldots, h-R_J$ образуют отрезок натурального ряда длины, не меньшей $R_P+1$, поэтому там есть все остатки). Давайте он сделает остаток 1, а после этого запретит себе ходить больше, чем на $R_P$. Даже после этого ограничения Пол всё равно проиграет, так как это ситуация с одинаковыми $r$, в которой $h$ делится на $r+1$, поэтому игрок, чья сейчас очередь ходить (то есть Пол), проигрывает.

Пусть $R_P > R_J$. Тогда, как бы ни походил Джон, останется больше одного яблока. В таком случае Пол поступит абсолютно аналогично: возьмёт ровно столько яблок, чтобы у оставшихся в миске был остаток 1 при делении на $R_J+1$, а потом запретит себе ходить больше, чем на $R_J$. Тогда Джон окажется в проигрышной ситуации игры с одинаковыми наибольшими силами крика, и Пол победит.

Итак, чтобы решить задачу, надо сначала рассмотреть случай $h \leqslant R_J + 1$ (при $h = 1$ побеждает Пол, а иначе — Джон), рассмотреть случай $R_J = R_P$ (при $h$, дающем остаток 1 при делении на $R_J$, побеждал Пол, а иначе — Джон), а в остальных случаях выигрывает тот, у кого больше его число $R$. Более того, из решения ясно, как следует ходить Джону, если он выигрывает: в таком случае всегда ход $t$, после которого $h-t$ имеет остаток 1 при делении на $R_P+1$, существует и является выигрышным.

# Разбор задачи «Сбор свидетельских показаний»

| | |
|---|---|
| Автор задачи: | Михаил Иванов |
| Подготовка тестов и решений: | Михаил Иванов, Арсений Кириллов |
| Автор разбора: | Михаил Иванов |

Для каждого свидетеля заведём список из него и всех знакомых с ним людей, пусть у $i$-го свидетеля список $l_i$. Заведём массив $m$ булевских значений (`true`/`false`) с индексами от 1 до $n$, означающий, получили ли мы уже свидетельские показания $i$-го человека (возможно, от одного из его знакомых) или нет. Изначально массив заполнен значениями `false`. Заведём целую переменную $k$, показывающую количество значений `false` в массиве, изначально равную $n$.

Будем перебирать людей одного за другим. Рассматривая $i$-го человека, мы для каждого человека $s$ из списка $l_i$ заменим $m[s]$ на `true`, поскольку, допросив $i$-го человека, мы получим показания $s$. Если при этом $m[s]$ действительно изменилось (то есть было равно `false` до этого изменения), то уменьшим на единицу $k$.

В конце рассмотрения $i$-го человека посмотрим, чему равно $k$. Как только впервые оно окажется нулевым, это будет значить, что допрос людей с первого по $i$-го дал нам все свидетельские показания, и можно заканчивать процесс и выводить ответ $i$. Так как $i \in l_i$, рано или поздно $k$ окажется равным нулю, хотя бы после рассмотрения всех $n$ свидетелей.

Асимптотика решения — $\mathcal{O}(m+n)$.

# Разбор задачи «Обеденное время»

| | |
|---|---|
| Автор задачи: | Даниил Орешников |
| Подготовка тестов и решений: | Даниил Орешников |
| Автор разбора: | Арсений Кириллов |

Заметим, что если текущее число минут не больше, чем $a$, то можно открутить время назад до 0 минут и тут же уйти на обед. Иначе нам не имеет смысла откручивать время назад, потому что это только увеличит время до следующего часа. Пусть $m$ — число минут. Если $m + b < 60$, то мы не можем получить следующий час на часах, и поэтому правильным ответом будет подвинуть часы на $b$ минут вперёд. Иначе же мы можем получить целое число минут, просто переведя время на начало следующего часа, поэтому правильный ответ — вывести следующий час и 0 минут. Но есть исключение: если сейчас уже 23 часа, то следующий час начнётся на следующий день, поэтому вывести необхрдимо `23:59`.

# Разбор задачи «Часовщик»

| | |
|---|---|
| Автор задачи: | Михаил Иванов |
| Подготовка тестов и решений: | Макар Селиванов |
| Автор разбора: | Макар Селиванов, Михаил Иванов |

Заметим, что необязательно помнить все стрелки. Достаточно оставить наборы $S_{\text{hour}}$, $S_{\text{minute}}$, $S_{\text{second}}$, состоящие из трёх самых тонких часовых, трёх самых тонких минутных и трёх самых тонких секундных стрелок соответственно (если стрелок какого-то типа менее трёх, мы их все добавляем в набор). Если какая-то стрелка является одной из самых тонких одновременно в нескольких категориях (например, самая тонкая секундная является второй по тонкости минутной), то мы её запишем по разу во все нужные наборы. Заметим, что для получения ответа достаточно рассматривать тройки стрелок из наших наборов: действительно, если в оптимальном ответе фигурирует стрелка какого-то типа, взятая не из набора этого типа (не умаляя общности, часовая стрелка не из $S_{\text{hour}}$), то её можно заменить на любую незанятую из часовых стрелок набора $S_{\text{hour}}$ (их три штуки, поэтому есть незанятая), при этом не ухудшив ответ. Дальше можно просто выбрать самую тонкую комбинацию, перебрав все возможные тройки из часовой, минутной и секундной стрелок соответственно из $S_{\text{hour}}$, $S_{\text{minute}}$, $S_{\text{second}}$ (тройку надо учитывать, только если все эти три стрелки различны).

# Разбор задачи «Треугольные числа»

| | |
|---|---|
| Автор задачи: | Даниил Орешников |
| Подготовка тестов и решений: | Даниил Орешников |
| Автор разбора: | Арсений Кириллов |

Заметим, что числа $a < b < c$ являются сторонами треугольника тогда и только тогда, когда $a+b > c$. Поэтому от набора необходимо и достаточно потребовать, чтобы сумма двух минимальных чисел в нём была больше максимального числа — ведь ясно, что, во-первых, $a + b$ всегда не меньше суммы двух минимальных чисел, а $c$ всегда не больше наибольшего числа, во-вторых, сами два минимальных и максимальное число должны образовывать длины сторон треугольника.

Примером набора, в котором сумма двух минимумов больше максимума, может служить любой отрезок натурального ряда, в котором первый член не меньше $n$. Например, подойдут $n + 1, n + 2, \ldots, 2n$ или, что даже проще придумать, $10^9 - n + 1, 10^9 - n + 2, \ldots, 10^9$.

# Разбор задачи «Объявления»

| | |
|---|---|
| Автор задачи: | Михаил Анопренко |
| Подготовка тестов и решений: | Михаил Анопренко, Макар Селиванов |
| Автор разбора: | Макар Селиванов |

В этой задаче надо просто посчитать количество букв в каждой возможной фразе и вывести минимальное количество букв. То есть, если станция встречается в списке остановок, то мы к длине второй фразы добавляем размер названия станции, а если не входит — то к размеру третьей. Заметим, что нам не важно, что названия станций могут повторяться, так как нам важно только количество букв, а порядок станций никак на него не влияет. Если количество остановок равно количеству станций между начальной и конечной, то появляется возможность вместо объявления всех станцию просто сказать первую фразу, однако это не всегда выгодно: например, если есть только две станции «`a`» и «`b`» и поезд на всех них останавливается, то выгоднее сказать «`Train goes with stops a, b`», что выгоднее на одну букву, чем «`Train goes with all stops`». Так что в этом случае надо вывести минимальное из двух возможных количеств букв.