Разбор задачи «Интересная характеристика»

Пусть $A > 0$. Рассмотрим битовую запись числа A и обозначим его старший единичный бит как x.

Докажем от противного, что ответа не существует. Пусть существует такое B, что $A \oplus B = A \& B$. Тогда, если x-й бит B равен 1, то x-й бит $A \oplus B$ равен 0, а x-й бит $A \& B$ равен 1. Аналогично, если x-й бит B равен 0, то x-й бит $A \oplus B$ равен 1, в то время как x-й бит $A \& B$ равняется 0. Это доказывает, что такого B не существует.

Разбор задачи «Ошибка министерства»

Воспользуемся тем фактом, что при равной площади фигур, периметр меньше у той, где стороны максимально похожи, то есть фигура приближенная к квадрату.

Тогда лучшим ответом будет квадрат, с площадью равным суммарной площадью доминошек.

Давайте найдём минимальный квадрат, в который гарантированно поместятся все доминошки. Обозначим его сторону за a, будем выкладывать доминошки типа 2 в горизонтально друг на друга, пока не забьём весь столбик и т.д. Очевидно, что при такой раскладке можно будет уместить $a^2 - 2 \times m$ доминошек первого типа. Отсюда можно увидеть, что квадрата со стороной $\sqrt{n + 2 \times m}$ хватит для покрытия всех доминошек, но стоит отметить, что необходимо целое число, поэтому надо брать сторону с округлением вверх: $\lfloor \sqrt{(n + 2 \times m} \rfloor + 1$

Таким образом, минимальный фигура — это квадрат со стороной $\sqrt{n + 2 \times m}$, а максимальная — это квадрат со стороной $\lfloor \sqrt{(n + 2 \times m} \rfloor + 1$. Между этими фигурами существует всего одна, которая может быть также решением — прямоугольник со сторонами $\sqrt{n + 2 \times m}$ и $\lfloor \sqrt{(n + 2 \times m} \rfloor + 1$.

Значит надо проверить все 3 варианта, а затем выбрать наилучший. Проверять определённую фигуру можно, например, сравнивая её площадь с суммарной площадью доминошек.

Разбор задачи «Погрызенный массив»

Решение на 40 баллов

Пусть $dp(l, r)$ означает, можно ли некоторой последовательностью укусов оставить лишь подотрезок $[l..r]$. Несложно видеть, что $dp(l, r)$ можно получить одним из двух видов укусов — либо отгрызть a слева и b справа, либо наоборот. То есть $dp(l, r)$ можно получить либо через $dp(l - a, r + b)$, либо через $dp(l - b, r + a)$. Такое решение легко реализовать за $\mathcal{O}(n^2)$.

Решение на 60 баллов

Суммарное количество укусов не превосходит $d = \lfloor \frac{n-1}{a+b} \rfloor$. Заметим, что после ровно i укусов у нас может получиться $i+1$ различных подотрезков (это следует из того, что наша конфигурация зависит только от количества дней, в которые Гриша откусывал по a элементов слева). Следовательно, можно просуммировать эти значения за $\mathcal{O}(n)$. Следует помнить, что случай $a = b$ необходимо рассмотреть отдельно.

Решение на 100 баллов

Свернем сумму из предыдущего пункта. $\sum_{i=0}^{d} i = (d + 1) \cdot (d + 2)$, что означает, что ответ можно получить за $\mathcal{O}(1)$.

Разбор задачи «Скоростная печать»

Рассмотрим тройку k, s, t для какого-то запроса. Пусть наибольший общий префикс s и t равняется x. Тогда оптимально вначале потратить $|s| - x$ миллисекунд на удаление неправильных символов, а затем $|t| - x$ миллисекунд на добавление нужных.

Ответ будет являться положительным тогда и только тогда, когда $|s| + |t| - 2 \cdot x$ совпадает с k по модулю 2. Почему это так? Несложно видеть, что мы всегда можем провести две миллисекунды впустую, добавив и удалив произвольный символ. В то же время четность ответа поменять не получится, так как смена четности подразумевает смену четности количества символов на экране.

Разбор задачи «Длинная дорога»

Рассмотрим граф на городах, где мы проводим рёбра только тогда, когда между городами есть путь стоимостью в 1 руританский доллар (неважно, в какую сторону). Нетрудно видеть, что все такие автобусы у нас есть. Если мы сможем найти в таком графе путь между первой и последней вершинами, то его (то есть, путь) можно вывести в качестве ответа.

Утверждение следующее: если $n \geqslant 5$, то в случае отсутствия такого пути правильным ответом будет -1. Действительно, предположим обратное: пусть есть путь, по которому мы точно сможем добраться от вершины 1 до вершины n. Тогда на нём есть ребро, не входящее в ранее построенный нами граф. Покажем, что существует граф без этого ребра, подходящий под условие задачи (а именно: покажем, что в полном графе без этого ребра можно построить все требующиеся пути).

Пусть требуемый путь между вершинами i и j имеет длину $a > 0$ (несложно видеть, что если $i \neq j$, то $a < n$, иначе $a > 2$ и $a \leqslant n$), а мы удалили ребро между вершинами l и m ($l \neq m$). Разберём несколько случаев:

1. Ни одно из чисел i, j не совпадает с числами l и m.

   В таком случае, если $a = 1$, пройдём по ребру между i и j. Если $a = 2$, то пройдём по пути i – l – j. Если $a = 3$, то из i перейдём в l, оттуда — в какую-либо из отличных от уже названных вершину (так как $n \geqslant 5$, то такая вершина существует), а оттуда в j. Если же $a > 3$, то из i перейдём в l, оттуда — в какую-либо из отличных от уже названных вершину, оттуда в m, после чего выйдем в оставшийся полный граф и походим там столько, сколько требуется для получения пути требуемой длины.

2. Какое-либо из чисел l, m не совпадает ни с i, ни с j

   Будем считать, что i совпадает с кем-то из чисел l, m. Опять же, если $a = 1$, то мы можем просто соединить i и j. Если $a = 2$, то проведём ребро из i в любую из вершин, отличных от i, j, k, l (такие есть, так как $n \geqslant 5$), оттуда — в j. Если же $a > 2$, то первое ребро проведём так же, как и в случае $a = 2$, далее — в ту из вершин i, j, которая не совпадает ни с i, ни с j, после чего выйдем в оставшийся полный граф и снова наберём там путь требуемой длины.

3. $i = l$, $j = m$ (или наоборот)

   Будем считать, что $i = l$, а $j = m$. Заметим, что случай $a = 1$ невозможен (иначе бы между l и m было ребро в графе, который мы строили первоначально, однако его там нет). В остальных же случаях достаточно походить по полному графу, который остался на отличных от начальной и конечной вершинах.

Нетрудно видеть, что разобраны все случаи.

Однако остаются ещё три варианта для различных n:

1. $n = 2$

   Самый простой случай: путь 1 – 2 всегда подходит из-за того, что нам гарантировали связность. Кроме того, нетрудно заметить, что в нём мы никогда не окажемся в ситуации, когда в первоначальном графе из первой в последнюю вершину попасть нельзя, поэтому его можно просто не разбирать.

2. $n = 3$

   Если расстояние между вершинами 1 и 3 не равно 1, то тогда оно равно 2, однако существует единственный такой простой путь: 1 – 2 – 3. Его и выведем.

3. $n = 4$

   Заметим, что если между какими-то двумя различными вершинами существует путь длины 3, то между двумя другими вершинами обязательно есть ребро-автобус (если вершины занумерованы числами a_1, a_2, a_3 и a_4, то между a_1 и a_4 существуют только два пути длины 3: $a_1 - a_2 - a_3 - a_4$ и $a_1 - a_3 - a_2 - a_4$, оба они используют ребро $a_2 - a_3$). Добавим все такие рёбра в наш граф. Если теперь нашёлся путь между вершинами 1 и 4, то выведем его.

   Утверждение: если путь не нашёлся, то ответом будет -1. Доказательство более-менее аналогично доказательству утверждения при $n \geqslant 5$.

Асимптотическая сложность решения: $O(n^2)$.

Разбор задачи «Конфеты за ответы»

Ключ к решению задачи на 100 баллов кроется в наблюдении, что максимальное количество делителей у чисел до $m = 10^6$ не превосходит $d_{max}(m) = 240$, причем равенство достигается на числе 720720.

Пусть $dp(n, x)$ — минимально возможное количество делителей у числа, в которое превращается n путем деления его на x простых (к примеру, $dp(6, 2) = 1$, так как можно последовательно поделить 6 на 2 и на 3, после чего получится число 1, у которого 1 делитель).

Введем следующие обозначения: скажем, что $g(x)$ — количество простых в разложении x (с повторами), а $d(x)$ — количество делителей x. Тогда

$$dp(n, x) = \min_{i|n,\, g(i)=x} d(\frac{n}{i}).$$

Это легко посчитать за $\mathcal{O}(m \log m)$.

Теперь поймем, как это знание нам пригодится. Будем обрабатывать запросы в оффлайне в порядке убывания k (от 240 до 1, так как ответ на любой запрос с $k > 240$ это 0). Для каждого индекса нашего массива будем поддерживать pos_i — минимальное значение такое, что $dp(a_i, pos_i) \leqslant k$. Несложно видеть, что pos_i может только увеличиваться. Вместе с тем, если мы знаем pos_i для текущего k, то любой запрос с таким k это просто сумма pos_i на отрезке $[l..r]$. Эту подзадачу проще всего решать префиксными суммами.

Оценим сложность второй части. Суммарное количество простых в разложении любого числа до m не превосходит $\log m$, следовательно, общее количество увеличений pos будет порядка $\mathcal{O}(n \log m)$. Количество же операций, связанных с вычислением префиксных сумм, будет равняться $\mathcal{O}(d_{max}(m) \cdot n)$. Каждый запрос будет обработан за $\mathcal{O}(1)$ через префиксные суммы.

Таким образом, итоговое время работы составит $\mathcal{O}((m + n) \log m + q + d_{max}(m) \cdot n)$. Стоит отметить, что можно было пропускать значения k, которые не соответствовали количествам делителей никаких чисел до миллиона, но для полного решения этого не требовалось.

Разбор задачи «Пушка и гора»

Для начала решим задачу без учета горы. Как мы знаем из курса физики, траектория полета снаряда с заданной начальной скоростью из точки (0,0) имеет вид:

$x(t) = (v \sin\alpha)t, y(t) = (v \cos\alpha)t - (gt^2)/2.$

Нам необходимо понять: существует ли такое α, что при каком-то $t \geqslant 0$ $x(t) = x_0$, а $y(t) = 0$.

Так как $g = 10$, получаем уравнение $(v \cos\alpha)t = (gt^2)/2 = 5t^2$, сократив на $t > 0$

$v \cos\alpha = 5t$, возведя в квадрат $v^2 \cos^2\alpha = 25t^2(1)$

$x_0 = x(t) = v \sin\alpha t$, а значит $x_0/t = v \sin\alpha$, возведя в квадрат $x_0^2/t^2 = v^2 \sin^2\alpha(2)$

Складывая (1) и (2) и домножая на t^2, получаем квадратное уравнение на t^2:

$v^2t^2 = x_0^2 + 5t^4$, Решая его, выражаем t и из (1) и (2) легко находим $\sin\alpha$, $\cos\alpha$. Все неизвестные в уравнении полета мы знаем. Тут важно понять, что может существовать **две** траектории.

Заметим, что если корней нет, значит попасть в точку невозможно.

А значит в первой подзадаче было достаточно сравнить дискриминант с нулем.

Также будет полезным выразить траекторию в виде $y(x)$

$y(x) = (\operatorname{ctg}\alpha)x - 5x^2/\sin^2\alpha$

Теперь решим вторую и третью подзадачу. Для начала заметим, что если x_0 левее левого подножья горы, то гора не влияет на решение, и все сводится к подзадаче выше. Иначе, заметим, что $y(x)$ это парабола с ветвями вниз, то есть вогнутая функция, а так как гора имеет вид треугольника, то снаряд попадает в гору тогда и только тогда, когда $y(x_v) < y_v$, где (x_v, y_v) координаты вершины горы.

Во второй подзадаче $(x_v, y_v) = (x_2, y_2)$

Третья же подзадача сводится к нахождению (x_v, y_v), а это стандартная задача пересечения двух прямых, каждая из которых задана парой точек. Её легко решить, например просто вспомнив формулу:

$$x_v = \frac{(x_1y_2 - y_1x_2)(x_3 - x_4) - (x_1 - x_2)(x_3y_4 - y_3x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

$$y_v = \frac{(x_1y_2 - y_1x_2)(y_3 - y_4) - (y_1 - y_2)(x_3y_4 - y_3x_4))}{((x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}$$

Разбор задачи «Знаки»

Решение на 20 баллов

Переберем все возможные расстановки знаков (их 2^{n-1}) и для каждой из них проверим, чему равен остаток по модулю 10. Это работает за $\mathcal{O}(n \cdot 2^n)$.

Решение на 70 баллов

Пусть $dp(i, x)$ — число способов расставить знаки между первыми i числами, чтобы выражение давало остаток x по модулю $p = 10$. Переберем позицию последнего плюса, тогда все, что правее, превратится в произведение. Иными словами (здесь и далее внутри индексов операции проводятся по модулю 10):

$$dp(i, x) = \sum_{j=0}^{i-1} dp(j, x - \prod_{u=j+1}^{u=i} a_i)$$

Необходимо помнить, что произведение в правой части необходимо брать по модулю 10, а также учитывать переполнение ответа и брать его по модулю $10^9 + 7$. Такое решение уже работает за $\mathcal{O}(p \cdot n^2)$.

Решение на 100 баллов

Модифицируем решение из предыдущего пункта. Пусть теперь $dp(i, x, y)$ — количество способов расставить знаки среди первых i чисел так, чтобы выражение вплоть до последнего знака плюс давало остаток x по модулю 10, а произведение от последнего знака плюс до i давало остаток y по модулю 10. Это изменение измельчает нам переход, ведь теперь нам не требуется перебирать все возможные позиции начала произведения. Вместо этого достаточно рассмотреть всего два перехода:

- Если мы ставим знак $+$ перед $i + 1$ элементом, то произведение прибавляется к сумме и заменяется на a_{i+1}, то есть $dp(i, x, y) \to dp(i + 1, x + y, a_{i+1})$.
- Если же мы ставим знак $*$ перед $i + 1$ элементом, то a_{i+1} вносит свой вклад в произведение, то есть $dp(i, x, y) \to dp(i + 1, x, y \cdot a_{i+1})$.

Ответ получается через всевозможные пары $dp(n, x, y)$. Такая динамика работает за $\mathcal{O}(p^2 \cdot n)$.

Разбор задачи «Подсказка к замку»

Решение 1.

Запишем код в виде восьми цифр: $d_1 d_2 \ldots d_8$. Запомним следующие шесть значений: $(d_1 + d_2 + d_3) \bmod 10$, $(d_2 + d_3 + d_4) \bmod 10$, $\ldots$, $(d_6 + d_7 + d_8) \bmod 10$. То есть для каждых трёх позиций подряд запомним, чему равна сумма цифр на этих позициях по модулю 10.

Восстанавливать будем так: если из каких-то трёх подряд цифр две известны, то третью можно восстановить однозначно. Можно проверить, что, какие бы четыре цифры из восьми ни были известны, такая тройка всегда найдётся. Затем найдём ещё тройку, в которой известны две цифры, и восстановим оставшуюся цифру. И так далее, пока все цифры не будут восстановлены.

Решение 2.

Общий случай предыдущего решения — запомнить шесть каких-нибудь линейных комбинаций восьми заданных цифр. Формально мы умножим вектор из восьми десятичных цифр слева на матрицу 6×8, у всех чисел результата возьмём остаток от деления на 10, и получим вектор из шести десятичных цифр. Матрица 6×8 должна быть такой, чтобы исходные числа можно было восстановить однозначно. Например, для решения 1 матрица будет иметь такой вид:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Существуют, однако, и другие подходящие матрицы. Если все остальные части решения уже написаны, хорошую матрицу можно просто подобрать.

При восстановлении можно просто перебрать $10\,000$ возможных чисел. Такое восстановление следует производить аккуратно, чтобы уложиться в ограничения по времени.

Решение 3.

Выберем какое-нибудь число m, близкое к 10^6, но не превосходящее его. Рассмотрим код как десятичное число x и запомним остаток $x \bmod m$. Следует помнить, что нужно выводить ровно шесть цифр, даже если при этом в числе будут ведущие нули.

Чтобы восстановить исходное число, можно перебрать все x' с запомненным остатком — их будет чуть больше ста — и для каждого проверить, совпадают ли четыре известные цифры.

Если удачно выбрать число m, то для любого остатка z все различные x', удовлетворяющие равенству $x' \bmod m = z$, будут иметь отличия хотя бы в пяти цифрах, поэтому x всегда удастся восстановить однозначно. Наибольшее такое удачное число — это $m = 998\,437$. Можно выбрать и какое-нибудь другое число.

Разбор задачи «Числовая пирамида»

Частичное решение для первой подзадачи ($n = 2$): если в треугольнике из соседних чисел неизвестно одно число, его легко восстановить.

К сожалению, такой подход не позволяет решить даже вторую подзадачу, например, при следующих входных и выходных данных:

$$\begin{array}{ccccc} & & 9 & & \\ & 0 & & 0 & \\ 8 & & 0 & & 9 \end{array} \quad \longrightarrow \quad \begin{array}{ccccc} & & 9 & & \\ & 4 & & 5 & \\ 8 & & -4 & & 9 \end{array}$$

Можно рассмотреть этот единственный особый случай для $n = 3$ отдельно — а ещё не забыть, что неизвестные числа могут оказаться отрицательными или даже нулями — и получить решение второй подзадачи.

Чтобы решить задачу целиком, посмотрим на разобранный выше пример более внимательно. Как мы узнали числа в нём? Например, мы могли назвать среднее число нижней строки буквой x и получить следующую пирамиду:

$$\begin{array}{ccccc} & & 9 & & \\ & x+8 & & x+9 & \\ 8 & & x & & 9 \end{array}$$

Получается, что $2x + 8 + 9 = 9$, откуда $x = -4$, и далее всё восстанавливается.

Оказывается, это же можно сделать и в общем случае. Например, выразим все числа пирамиды размера $n = 5$ через числа в нижней строке:

$$\begin{array}{ccccccccc} & & & & a+4b+6c+4d+e & & & & \\ & & & a+3b+3c+d & & b+3c+3d+e & & & \\ & & a+2b+c & & b+2c+d & & c+2d+e & & \\ & a+b & & b+c & & c+d & & d+e & \\ a & & b & & c & & d & & e \end{array}$$

Получается, что заданные числа задают линейные уравнения из элементов нижней строки. В качестве примера слева показана пирамида, а справа — система линейных уравнений, которую она задаёт.

$$\begin{array}{ccccccccc} & & & & 0 & & & & \\ & & & 0 & & 8 & & & \\ & & 0 & & 4 & & 0 & & \\ & 2 & & 0 & & 0 & & 2 & \\ 0 & & 1 & & 1 & & 0 & & 1 \end{array} \longrightarrow \begin{cases} b = 1 \\ c = 1 \\ e = 1 \\ a + b = 2 \\ d + e = 2 \\ b + 2c + d = 4 \\ b + 3c + 3d + e = 8 \end{cases}$$

Как известно, систему линейных уравнений можно решать методом Гаусса. В задаче гарантируется, что решение существует, единственно и состоит из целых чисел. Значит, решим систему методом Гаусса в вещественных числах (при этом может появиться небольшая погрешность), после чего округлим каждый ответ до ближайшего целого числа. Наконец, восстановим по нижней строке всю числовую пирамиду.

Разбор задачи «Простая задача»

Известно, что k-е простое число примерно равняется $k \log k$, где $\log k$ означает *натуральный* логарифм числа k.

Решение на 7 баллов

Здесь было достаточно последовательно проверять числа на простоту. Примерное время работы можно оценить как $\mathcal{O}(k \log k \sqrt{k \log k})$.

Решение на 16 баллов

Для решения второй подзадачи было необходимо использовать *решето Эратосфена*. Если грамотно оценить верхнюю границу отрезка для решета как $C \cdot k \log k$, то время работы составит $\mathcal{O}(k \log^2 k)$.

Решение на 27-54 баллов

Заменим решето Эратосфена на *линейное решето*. Время работы упадет до $\mathcal{O}(k \log k)$. В зависимости от того, насколько эффективна используемая реализация, количество баллов может варьироваться от 27 до 54.

Решение на 69 баллов

При некоторой доле везения, хорошем коде и использовании *блочного решета* можно было достичь отметки в 69 баллов. Впрочем, это был максимум для решений, использующих решето.

Решение на 100 баллов

Представим на секунду, что мы точно знаем, что некоторое простое число p является x-м по счету простым числом. Тогда, если $x \leqslant k$, разумно исключить из рассмотрения числа, меньшие p, так как среди них точно нет ответа.

Предподсчитаем локально все простые числа до $2 \cdot 10^9$, и запомним из них каждое m-е (например, при $m = 3 \cdot 10^4$ запомненных чисел будет всего 3300). Тогда длина промежутка между двумя последовательными запомненными числами будет приблизительно равняться $20 \cdot m$. Соответственно, алгоритм поиска k-го простого числа выглядит так: сначала мы находим приближение по формуле $p' = memo(\lfloor \frac{k}{m} \rfloor)$, после чего последовательно проверяем числа на простоту, пока не найдем искомое. Стадия проверки может быть реализован и более быстрым способом (например, блочным решетом из предыдущего пункта).