Разбор задачи «Оно»

Заметим, что если какой-то отрезок входит в ответ, любой его подотрезок тоже входит в ответ. Обозначим за r_i максимальную правую границу отрезка строки s, начинающегося с позиции i, который можно получить из символов строки t. Несложно заметить, что $r_i \leqslant r_{i+1}$. Найдем r_0, для этого, посчитаем количество вхождений каждого символа в строку t, и будем идти по строке s слева направо, уменьшая счетчики встреченных символов, пока счетчик одного из символов не станет отрицательным. Теперь, что того, чтобы найти r_1, нужно передвинуть границу рассматриваемого отрезка вправо на 1, для этого просто увеличим счетчик, соответствующий символу s_0, и снова будем двигать правую границу, пока счетчики неотрицательны. Будем продолжать так делать для всех i. Итоговое время работы — $O(n)$.

Разбор задачи «Деревянный замок»

Заметим, что существует оптимальное решение, в котором в первую очередь применяются все операции перекрашивания, а потом все операции удаления. В таком случае, ответом является количество перекрашиваний плюс количество одноцветных компонент связности после перекраски.

Обозначим за 0 белый цвет, а за 1 — черный. Подвесим дерево за любую вершину. Посчитаем $dp[i][j]$ — минимальное количество операций, необходимое, чтобы удалить все поддерево вершины i, если после перекрашиваний и до удалений она будет иметь цвет j. Тогда $dp[i][j] = \sum\limits_u \min(dp[u][j] - 1, dp[u][j \oplus 1]) + cost_{i,j} + 1$, где вершина u — сын вершины i в дереве, а $cost_{i,j}$ — стоимость покраски вершины i в цвет j (0, если вершина изначально имеет цвет j, и 1 иначе). В сумме для каждого сына u вершины i мы выбираем:

- Покрасить его в один цвет с вершиной i, тогда одноцветные компоненты связности для i и u объединятся, поэтому нужно вычесть 1
- Покрасить его в цвет противоположный цвету i

После этого, ответом на задачу будет являться $\min(dp[root][0], dp[root][1])$, где $root$ — номер вершины, являющейся корнем подвешенного дерева.

Разбор задачи «Алмазы»

Для каждой вершины существует не более чем $O(\sqrt{m})$ соседей большей степени.

Почему? Потому что если у вершины степень не более чем $O(\sqrt{m})$, у нее самой существует не более $O(\sqrt{m})$ соседей. Если же у вершины степень больше корня, то существует не больше чем $O(\frac{m}{\sqrt{m}})$ вершин степени больше корня, что есть $O(\sqrt{m})$.

Таким образом, можно решать задачу следующим образом:

Зафиксируем вершину v степени три в алмазе, запомним в массиве вершины, которы являются ее соседями. Затем переберем соседа u этой вершины и соседа вершины u, но у которого степень больше (при равенстве степеней можно сравнивать по номеру). Если этот сосед соединен с v (что вы проверяете за $O(1)$, так как вы запомнили в массив), то прибавим к степени вершины u и степени этого соседа единицу.

Таким образом, для всех соседей вершины v мы нашли, сколько у них есть соседей, которые также являются соседями вершины v. Не трудно заметить, что количество алмазов, в которых вершина v имеет степень три равно $\sum \frac{deg_v(deg_v - 1)}{2}$.

В конце каждый алмаз мы посчитали два раза, поэтому нужно поделить ответ на два.

Время работы $O(m\sqrt{m})$ и $O(m)$ памяти.

Разбор задачи «Хорошее подмножество»

В данной задаче ограничения на числа $a_i \leqslant 10^{18}$.

В данных ограничениях задачу можно решать следующим образом:

Проверим как НОД все простые $\leqslant 10^6$. Для этого для каждого простого в этом интервале проверим, сколько чисел на него делится, разделим эти числа на максимальную степень этого простого, на которую он делится. И прорелаксируем ответ количеством.

Теперь все числа состоят не больше чем из двух различных простых множителей.

Далее утверждается, что как возможный НОД в ответе достаточно рассмотреть только попарные НОДы чисел.

Таким образом можно выделить все различные НОДы пар, и проверить количество чисел, которые на них делятся, прорелаксировав этими величинами ответ.

Авторское решение работает при больших ограничениях на числа, к примеру $a_i \leqslant 10^{36}$.

Для того чтобы решить задачу в таких ограничениях давайте поддерживать неразделимые «группы», такое множество попарно взаимно простых чисел, что каждое данное число можно представить в виде произведения чисел из этого множества.

Очевидно, в этом множестве не больше чем $C \cdot n$ чисел, где C — максимальное количество разных простых, которые есть в числах. В ограничениях $a_i \leqslant 10^{18}$, $C = 15$, очевидно, при $a_i \leqslant 10^{36}$, $C \leqslant 30$, но на самом деле меньше

Как найти эти группы? Давайте добавлять числа по одному и поддерживать это разбиение для текущего префикса чисел. При добавлении нового числа x, если найдется число y из множества, которое имеет НОД g больший, чем один, с данным числом, можно заменить это число y из группы на g, и попытаться добавить в группу числа $\frac{x}{g}$ и $\frac{y}{g}$. Если же в какой-то момент мы пытаемся добавить в группу число 1, следует выйти. Авторское решение производит добавления в группу с помощью рекурсивного алгоритма.

Таким образом за $O(C \cdot n^2)$ можно найти эти неразделимые группы, а затем для каждой группы посчитать количество чисел, которые делятся на это значение, и прорелаксировать этим ответом.

Разбор задачи «Защитный узор»

Задачу можно решить методом динамического программирования по изломанному профилю. Состоянием является разбиение черных клеток, находящихся в текущем профиле, на компоненты связности. При переходе нужно выбрать цвет текущей клетки из двух вариантов: белого и черного. Стоимость перехода равна 0, если клетка уже имеет такой цвет, и 1, если исходно цвет клетки противоположный. Состояние можно задавать массивом целых чисел длины m, в котором i-е число равно -1, если соответствующая клетка покрашена в белый цвет, и какому-то неотрицательному числу иначе. Причем, если две черные клетки находятся в одной компоненте связности, им соответствуют одинаковые числа, а иначе — разные.

При переходе, в котором текущая клетка красится в белый, нужно проверять, что каждая компонента связности все еще имеет представителя в следующем изломанном профиле. Это может нарушиться только если соседняя сверху клетка — черная, и является единственным представителем своей компоненты связности. При переходе, в котором текущая клетка красится в черный, нужно проверять, что не образовался цикл по черным клеткам. Это может произойти только в том случае, если соседние клетки слева и сверху являются черными и лежат в одной компоненте связности.

Если переход корректен, нужно построить новый массив, который соответствует текущему разбиению клеток в изломанном профиле на компоненты связности. Если всегда, идя слева направо по профилю, выбирать в качестве номера очередной компоненты минимальное неиспользованное целое неотрицательное число, максимальное количество различных достижимых состояний для $m = 10$ оказывается $\sim 7\,500$. Итоговое время работы равно $O(n \cdot m \cdot S \cdot m)$, где $S \approx 7\,500$ — число различных состояний.

Разбор задачи «Они»

Рассмотрим $\texttt{pref}_i$ — сумму чисел с индексами от 1 до i включительно и $\texttt{suf}_j$ — суммму чисел с индексами от j до n включительно. Тогда мы ищем минимум $|pref_i - suf_j|$ по всем $i < j$.

Заметим, что как $\texttt{pref}_i$ возрастает при возрастании i, так и $\texttt{suf}_j$ возрастает при убывании j, значит можно решить задачу двумя указателями по монотонным функциям `suf` и `pref`.

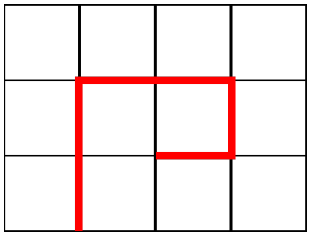
Заметим, что и `suf`, и `pref`, можно считать за $\mathcal{O}(1)$, предподсчитав префиксные или суффиксные суммы. А затем идти двумя указателями, при $\texttt{pref}_i > \texttt{suf}_j$ уменьшая j, а при обратном неравенстве увеличивая i. При равенстве мы получаем ответ 0. А сам перебор заканчивается как только выполняется условие $i \geqslant j$, так как после этого момента рассматриваемые суффиксы и префиксы нарушают условие задачи.

Очевидно, что таким проходом мы рассмотрим все минимальные разности между префиксами и суффиксами. А если в какой-то момент суффикс оказался равен префиксу, то минимальная разность достигается именно в этом месте и равна 0. Таким образом, задача решается за $\mathcal{O}(n)$.

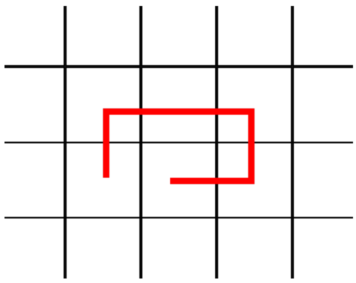
Разбор задачи «Ножницы»

Решением данной задачи для чисел n и m является довольно простая формула. А именно $(n-1) \times (m-1)$. Есть множество способов прийти именно к такому решению, но мы разберем самый наглядный.

Для начала вспомним рисунок из условия:



Затем сделаем следующее преобразование: заменим все клетки на границы, а границы на клетки таким образом, чтобы смежность сохранилась. Тогда для рисунка из условия преобразование выглядит следующим образом:



Теперь заметим, что в новом прямоугольнике нужно спиралью покрыть все клетки, кроме граничных (они отвечали за границы изначального поля и там резать было нельзя). На последнем рисунком красным отмечен маршрут. Очевидно, что мы сможем посетить все клетки, а также, что длина разреза оригинального поля совпадает с числом таких клеток, соответственно равна $(n-1) \times (m-1)$.

Разбор задачи «Побег из заброшенного дома»

Пусть x_s и y_s — номер строки и столбца клетки, в которой находится символ «`s`», а x_f и y_f — тоже самое дли символа «`f`». Сначала, с помощью dfs проверим, достижимы ли вообще друг из друга эти клетки. Если они достижимы, то мы знаем чётность шагов по горизонтали, которые мы сделаем. Оно совпадает с чётностью числа $\left|x_s - x_f\right|$. Аналогично, мы знаем чётность числа шагов по вертикали. Тогда если из стартовой вершины достижима вершина, из которой можно сделать ход по горизонтали, и достижима вершина, из который можно сделать ход по горизонтали, то с помощью них мы можем получить любую возможную разность между ходами по горизонтали и вертикали, которые удовлетворяют условиям чётности ходов. Тогда ответом будет чётность числа $\left|x_s - x_f\right| + \left|y_s - y_f\right|$.

Если же какая-то из этих вершин не достижима, то мы можем ходить либо в пределах одной строки, либо в пределах одного столбца, и тогда ответом является $\left|x_s - x_f\right| + \left|y_s - y_f\right|$.

Разбор задачи «Звуки в подвале»

Если в исходной полоске цвет первой клетки совпадает с цветом последней клетки, Билл проигрывает, потому что не может сделать первый ход.

А если цвет первой и последней клеток исходной полоски различаются, Билл всегда может разрезать полоску так, что обе получившиеся полоски будут иметь концы одинаковых цветов. Например, он может отрезать максимальный по длине одноцветный префикс полоски. Он может сделать такой ход, потому что полоска содержит оба цвета и значит такой префикс не совпадает со всей полоской. После этого, цвета концов левой части совпадают, потому что Билл отрезал одноцветный префикс. А цвета концов правой части совпадают, потому что левый конец правой части имеет цвет отличный от цвета левого конца исходной полоски, а значит равный правому концу исходной строки и правой части.

Поэтому, во втором случае Билл выигрывает.

Разбор задачи «Перестроения»

Заметим, что для решения задачи достаточно получать из тождественной перестановки $\{1, 2, \ldots, n\}$ произвольную перестановку. Получение перестановки b из перестановки a эквивалентно получению перестановки $a^{-1}b$ из тождественной перестановки, где a^{-1} — обратная к a перестановка.

Пусть мы хотим получить из тождественной перестановки перестановку p. Будем решать задачу рекурсивно. Рассмотрим последнее перестроение, после которого мы получим перестановку p. Найдем решение, при котором в этом перестроении из ряда вышло $k = \lfloor \frac{n}{2} \rfloor$ человек. Очевидно, это люди с номерами $p_1, p_2, \ldots, p_k$, так как в конце именно они должны стоять в начале ряда. Разобьем всех людей на две группы: людей с номерами $p_1, p_2, \ldots, p_{k-1}, p_k$ и людей с номерами $p_{k+1}, p_{k+2}, \ldots, p_{n-1}, p_n$. Для решения задачи необходимо и достаточно, чтобы перед последним перестроением люди из первой группы стояли друг относительно друга в порядке $p_k, p_{k-1}, \ldots, p_2, p_1$ (ведь при последнем перестроении их относительный порядок изменится на обратный), а люди из второй группы стояли друг относительно друга в порядке $p_{k+1}, p_{k+1}, \ldots, p_{n-1}, p_n$. При этом относительный порядок людей из разных групп нам не важен.

Заметим, что теперь задача разбилась на две независимых: для людей внутри каждой из групп нам нужно из некоторого исходного их порядка получить другой порядок. Заметим, что эти задачи можно решать параллельно, то есть с помощью одного перестроения всего ряда совершать одновременно произвольное перестроение внутри первой группы и произвольное перестроение внутри второй группы. Так как относительный порядок людей из разных групп нам не важен, результат от этого не изменится.

Значит, задачу можно решать рекурсивно. Построим необходимые перестроения для каждой из групп, а после этого объединим пары перестроений внутри групп в перестроения всего ряда.

Мы свели задачу для перестановки размера n к двум задачам для перестановок размера $\frac{n}{2}$ и одному дополнительному перестроению. Значит, общее количество перестроений для перестановки размера n не превысит $\lceil \log_2 n \rceil$. Этого хватает, чтобы уложиться в 15 перестроений для перестановок размера не более 10 000.