Разбор задач TopSpeedCoder

Задача A: Гарри Поттер и три заклинания

Если a, b, c, d, e, $f > 0$, то нам просто нужно проверить, что $\frac{bdf}{ace} > 1$. Остается аккуратно разобрать случаи, когда какие-то параметры равны 0.

Задача B: Антисортировка

В этой задаче достаточно отсортировать массив и поменять первые 2 элемента местами. Так как $n \geq 3$, то после этого мы получим неотсортированную последовательность.

Задача C: Астрид и квадраты

Будем честно эмулировать разрезания. Получим следующую формулу:

$$f(n, m) = \left\lfloor \frac{n}{m} \right\rfloor + f(n \bmod m, m)$$

(эта формула работает если $n \geq m$). Если честно ее вычислять, то время работы будет логарифмическое(по аналогии с алгоритмом Евклида).

Задача D: Берт и землеройки

Максимального времени можно добиться, если группа будет преобразовываться так: $n \to (n-1, 1) \to n-2$. Тогда ответом будет $\left\lfloor \frac{n+1}{2} \right\rfloor$.

Минимального времени можно добиться, если группа будет преобразовываться так: $n \to (\left\lfloor \frac{n}{2} \right\rfloor, \left\lceil \frac{n}{2} \right\rceil) \to (\left\lfloor \frac{n}{2} \right\rfloor - 1, \left\lceil \frac{n}{2} \right\rceil - 1)$. Тогда ответом будет $T(n) = 1 + T(\left\lceil \frac{n}{2} \right\rceil - 1) = \lfloor \log_2(n+1) \rfloor$.

Задача E: Марсианская архитектура

Для того, чтобы добавить к подотрезку $[l, r]$ арифметическую прогрессию можно посмотреть на разности между соседними элементами и посмотреть в каких точках эти разности меняются. В точке l делается $+1$ к разности, в точке $r+1$ делается -1 к разности. Тогда мы просто идем слева направо по массиву и накапливаем текущую разность(это на самом деле аналог производной) и вычисляем значение в точке через прошлое и текущую разность.

Задача F: Камилла и язык WR

В данной задаче нужно честно проэмулировать выполнения программы на языке WR.

Задача G: Ворливые коровы

Рассмотрим стартовую перестановку коров. Выделим в ней циклы. Рассмотрим один цикл длины L и попытаемся его отсортировать за минимальную стоймость. Пусть стоймость i-й коровы равна C_i. Понятно, что мы должны сделать хотя бы $L-1$ обмен. Рассмотрим два элемента в цикле. Тогда если мы их поменяем местами, то цикл распадется на два цикла длинами L_1 и L_2, такие что $L_1 + L_2 = L$. Заметим, что таким образом каждый элемент нам придется хотя бы раз с кем-то поменять. Тогда давайте делать так: меняем местами минимальный элемент на цикле с его соседом так, чтобы остался цикл длины $L-1$, и платим мы $C_{min}+C_i$. Таким образом мы заплатим минимальную стоимость$((\sum_{i=1}^{L}(C_i+C_{min}))-2C_{min})$, чтобы отсортировать перестановку, состоящую из одного цикла. Но теперь вспомним, что у нас есть еще и другие циклы. Заметим, что если у нас есть два цикла, то при перемене элементов из этих циклов мы получим один большой цикл, причем мы это можем сделать за сумму минимальных элементов на циклах. Тогда получаем жадное решение общей задачи - найти все циклы, затем выделить цикл с самым минимальным элементом. Теперь для каждого цикла мы смотрим что выгоднее - развернуть его описанным выше способом или приклеить его к циклу с минимальным элементом, а потом развернуть в конце весь минимальный цикл.

Задача H: Давид и осёл

Будем идти снизу вверх — в последнем ряду слева-направо, в предпоследнем справа-налево, в предпредпоследнем опять слева направо, итд.

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

Задача I: Эмма и соты

Можно просто построить многоугольник по обходу, вычислить его площадь и поделить на площадь одного шестиугольника.

Задача J: Побег

Будем эмулировать действия принессы и дракона. Считаем время, нужное для того, чтобы дракон долетел до принцессы. В этот момент принцесса скидывает драгоценность, дракон долетает до своей пещеры. Повторяем процесс пока принцесса не добежит до замка.

Задача K: Петя и Java

Проще всего было считать число из инпута на Java в типе BigInteger и честно сравнить с заданными из условия границами. Если писать решение на C++, то можно было считать числа как строки и сравнивать их сначала по длинне, а при равенстве длин как строки.

Задача L: Цепная дробь

Пусть мы раскладываем в цепную дробь число вида $\frac{a\sqrt{n}+b}{c}$, где a, b, c целые числа, $c>0$. Изначально $a=1$, $b=0$, $c=1$. Честно вычислим $a_0=\left\lfloor\frac{a\sqrt{n}+b}{c}\right\rfloor$. Теперь нам нужно разложить в цепную дробь число

$$\frac{1}{\frac{a\sqrt{n}+b}{c}-a_0}=\frac{c}{a\sqrt{n}+b-a_0c}=\frac{c(-a\sqrt{n}+b-a_0c)}{(a\sqrt{n}+b-a_0c)(-a\sqrt{n}+b-a_0c)}=\frac{-ac\sqrt{n}+(bc-a_0c^2))}{(b-a_0c)^2-n}$$

Мы можем сократить числитель и знаменатель на $\gcd(a_{new},b_{new},c_{new})$ и рекурсивно запуститься. Заметим, что если начинать с $(1,0,1)$, то все a будут равны 1, а $|b|,|c|\leqslant 2\sqrt{n}$. А длина цикла ведет себя как $O(\sqrt{n}\cdot\ln n)$. Можно попробовать доказать эти факты самим или найти их в литературе по цепным дробям.

Задача M: Шестеренки

Эта задача является простым вариантом китайской теоремы об остатках. Из нее следует, что если существует ответ, то он не превосходит наименьшего общего кратного модулей. Это означает, что можно просто перебрать все варианты и проверить.

Задача N: НОК

Понятно, что ответ равен $n(n-1)(n-2)$, если $gcd(n, n-2) = 1$. Из этого соображения следует, что в ответе a, b, $c \approx n$. Можно перебрать все тройки чисел в интервале $[n-10, n]$ и получить ОК, но можно написать полный перебор всех вариантов в порядке убывания с отсечениями по ответу, который будет работать примерно за $O(n)$ из-за того, что мы уже имеем отличное стартовое приближение к ответу.

Задача O: Сон студента

Нужно честно попробовать 2 варианта расположения мальчика и девочки. Они могут ходить вместе, если $d-1 \leqslant m$ и $m \leqslant 2(d+1)$ (m - число пальцев на руке мальчика, d - число пальчев на руках девочки).

Задача P: Постройка пирамиды

Можно просто по очереди снимать слои.

Задача Q: Треугольная комната

Можно просто перебрать три вершины и проверить длины сторон на совпадение с длинами из условия.

Задача R: SMS

Нужно аккуратно разбить текст на предложения и жадно распихать предложения по смскам.

Задача S: Неквадраты

Самое главное в этой задаче - понять второй семпл:

$$7 = -1 \times -7.$$

А дальше понятно, что если k четное, то можно сделать все множители отрицательными(в качестве множителей, например могут выступать само n и $k-1$ единица) и все будет хорошо. При нечетном k мы понимаем, что 1 никак нельзя представить требуемым образом. Оставшиеся числа можно предтавить так - взять любой простой делитель p числа n, $-\frac{n}{p}$ и $k-2$ минус единицы.

Задача T: Перестановка цифр

При прочтении условия сразу хочется написать жадный алгоритм перемены цифр местами, но к сожалению он неверен. Правильным решение является динамика $\mathrm{dp}[n][k]$ - можно ли получить число n за k шагов. Переход осуществляется за $\log_{10}^2(n)$ простым перебором того, что нужно менять местами. Так как мы всего лишь меняем цифры местами, то состояний у нас будет не nk, а $(\log_{10}(n))!k$ и можно их хранить не все, а только текущие достижимые состояния при фиксированном k.

Задача U: Пристрастный учитель

Раздадим изначально каждому ученику по одной конфете, а потом будем повторять следующую операцию пока процесс не стабилизируется: находим место, где не выпоняются условия и дадим конфеты ученикам так, чтобы неравенства выполнялись. Понятно, что суммарно мы дадим не более, чем n^2 конфет. Получаем очень простое решение за $O(n^3)$.

Альтернативное решение: пусть у первого ученика будет x конфет. Тогда мы можем вычислить число конфет у каждого из учеников. Останется только найти такое минимальное x, что у каждого будет хотя бы одна конфета.

Задача V: Поезда

В принципе понятно, что Вася будет чаще ездить к той девушке, к которой чаще ездят поезда. В нашем случае это почти так - сначала заметим, что если сократить a и b на одно и тоже число, то ответ не поменяется. Несложно доказать, что если a и b взаимно просты и отличаются на 1, то ответ Equal. Иначе действует наше изначальное предположение. Но в задаче можно было честно проемулировать процесс до НОК(a, b) за $O(a + b)$.

Задача W: Ловушки

Нужно просто посчитать у каждой вершины степень и выдать те вершины у которых степень равна 1, кроме корня.

Задача X: Магия вуду

Переберем за 2^n какие числа мы возьмем и честно вычислим максимум. Понятно, что нужно умножать и брать по модулю на каждом шаге, а не в конце, чтобы избежать работы с большими числами.