# Разбор задачи «Плащ левитации»

На первый взгляд становится понятно, что для решения этой задачи нужно проверить, что:

- либо $a/2 \leqslant h$ и $b \leqslant l$
- либо $b/2 \leqslant h$ и $a \leqslant l$

Однако, присутствует не очень очевидный случай, когда $a$ нечетно. В этом случае, если делить целочисленно, то можно допустить ошибку. Например, если $a = 5$, то при целочисленном делении $a/2 = 2$, и для $h = 2$ действительно будет выполнятся $a/2 \leqslant h$, однако касание пола в такой ситуации не избежать.

Таким образом при делении на два, нужно не забыть учесть случай нечетных размеров плаща.

# Разбор задачи «Граненые стаканы»

Нужно найти суммарную площадь многоугольников, и поделить объем на нее.

Площадь многоугольника можно найти с помощью векторных произведений.

# Разбор задачи «Преследование»

**Первое решение:**

Будем считать динамику $dp_{ijt}$ — количество разбиений отрезка числа $[0, j]$, если последнее выбранное число в нем — $[i, j]$, и уже выбрано $t$ чисел.

Переход: находясь в текущем состоянии $(i, j, t)$, переберем новое число в разбиении — переберем $p$ от $j+1$ до $|x|$ и попробуем взять число $x[j+1..p]$, если абсолютная разность с предыдущим попадает в отрезок $[l, r]$.

Итоговая асимптотика решения: $O(|x|^4)$, но длина числа $x$ меньше 20, поэтому это легко укладывается по времени даже со 100 мультитестами.

**Второе решение:**

Давайте заметим, что всего возможных разбиений $C_{18}^{9}$, что меньше 50000. Поэтому достаточно просто перебрать все возможные разбиения и проверить их на корректность.

# Разбор задачи «Библиотека»

Первоначально преобразуем данные о каждой книги в отрезки дней, в каждый из которых книга уже может быть сдана, то есть отрезки $books$ вида $[s_i + c_i, f_i]$ — от момента прочтения книги до дня, по прошествии которого книга должна быть сдана. Затем отсортируем все отрезки по левому концу.

Задача решается жадностью: в каждый свободный день выбираем из всех книг, которые к этому дню уже прочитаны, книгу, которая должна быть сдана раньше остальных. Затем переходим к следующему дню и проделываем то же самое.

Реализовывать будем следующим образом. Каждый правый конец отрезка по мере возможности сдачи соответствующей ему книги в текущий день будем добавлять в множество. Текущий день храним в переменной $j$, еще не рассмотренный отрезок - в переменной $i$. Первоначально $j$ соответствует минимальному левому концу среди левых концов всех отрезков. На каждой итерации цикла сначала добавляем все доступные для сдачи в текущий день $j$ книги (увеличивая при этом переменную $i$), а затем вынимаем из множества минимальный правый конец среди всех правых концов доступных книг. И если вынутый правый конец оказался меньше $j$, то есть текущего свободного дня, выводим `NO` и завершаем программу.

Иначе идем дальше. Если после извлечения элемента множество оказалось пусто, присваиваем переменной $j$ значение левого конца еще не рассмотренного отрезка, то есть $j = books[i].left$. В ином случае, если множество не оказалось пустым, увеличиваем $j$ на один. Затем переходим к следующей итерации.

После цикла выводим `YES`.

# Разбор задачи «Доктор Стрэндж и перестановка»

Нужно проверить, что в массиве есть два либо ноль элементов, четность которых не совпадает с индексами, на которых они стоят.

Если таких элементов два, то проверяем, что четности их разные, и меняем местами.

Если таких элементов ноль, то меняем элементы, стоящие на 1 и 3 позициях. Частным случаем является случай при $n = 2$, для него невозможно произвести обмен в этом случае.

# Разбор задачи «Послание»

Для того, чтобы максимизировать число строк $c$, будем жадно по порядку набирать символы строки $c$, встречающиеся в строке $s$, так как в строке $s$ эти символы так же идут по порядку.

# Разбор задачи «Знания — сила»

Для начала заметим, что носители размножаются независимо друг от друга. А значит, достаточно почитать ответ $ans$, если изначально был только один носитель, тогда итоговое количество — $ans \times n$. Все дальнейшее рассуждение предполагает, что изначально есть только один носитель.

Пусть $A_i$ — количество носителей, появившихся за $i$-ый дней. То есть

$A_1 = 2$, $A_2 = 5$, $A_3 = 13$ и так далее.

Тогда верно равенство

$A_n = A_{n-1} + \sum\limits_{i=1}^{n-2} A_i$, где $A_{n-1}$ — количество носителей, появившихся за $n-1$ день, $\sum\limits_{i=1}^{n-2} A_i$ — количество новых носителей.

Заметим, что $F_{2k} = A_k$. Докажем данное утверждение по индукции.

Предположим, что для 1 $lei\ lek$ равенство верно. Тогда докажем равенство для $k+1$. То есть мы хотим доказать, что

$F_{2(k+1)} = A_{k+1}$, тогда

$A_{k+1} = A_k + \sum\limits_{i=1}^{k-1} A_i = F_{2k} + F_{2k-1}$

Заметим, что $A_k = F_{2k}$ по индукционному предположению. Тогда должно соблюдаться, что

$\sum\limits_{i=1}^{k-1} A_i = F_{2k-1}$

Вновь распишем левую и правую части равенства, получим

$\sum\limits_{i=1}^{k-1} A_i = A_{k-1} + \sum\limits_{i=1}^{k-2} A_i = F_{2k-2} + F_{2k-3}$

Вновь заметим, что по индукционному предположению

$A_{k-1} = F_{2k-2}$. Тогда повторим процесс заново для оставшейся суммы, каждый раз индексы будут уменьшаться, и одно из слагаемых и в правой и в левой части будут сокращаться. В итоге придем к равенству

$A_1 = F_2$ — очевидно верно.

# Разбор задачи «Путешествие сквозь миры»

Различных цифр, из которых могут состоять номера, интересные для Стрэнджа, 9 штук. Переберем и зафиксируем цифру $d$, а затем посмотрим на все подходящие числа, состоящие только из этой цифры. Для этого будем последовательно рассматривать числа $\overline{d}$, $\overline{dd}$, ... (здесь под $\overline{x_1x_2x_3\ldots x_n}$ подразумевается десятичная запись числа $x_1x_2\ldots x_n$). Как только первое из них станет не меньше $l$, мы попали в отрезок $[l, r]$ и можно начинать обновлять ответ. Как только число стало больше $r$, дальше можно не продолжать.

Также нужно было быть аккуратным с переполнениями — при неаккуратной реализации могло не хватить даже 64-битного типа данных.

# Разбор задачи «Карточный трюк»

Рассмотрим простую эмуляцию того, что написано в программе. Такая программа может не уложиться в ограничение по времени.

Заметил, что некоторое количество раз подряд прямоугольник, образованный пересечением карт, будет одинаковый. Пусть $a \leqslant b$ и $c \leqslant d$, тогда до тех пор, пока $b \geqslant c$, $b \geqslant a$, $d \geqslant a$ и $d \geqslant c$, будут сохранятся равенства, что $a \leqslant b$ и $c \leqslant d$, поэтому ориентация прямоугольников меняться не будет, а пересечением прямоугольников будет являться прямоугольник размера $c \times a$. Можно вычислить количество раз, которое можно отрезать такой прямоугольник, чтобы выполнялись все 4 неравенства, это и нужно сделать. После этого изменить ориентацию прямоугольников. И повторять это, пока они не исчезнут.

Заметим, что этого также недостаточно. Например, если одна из карт имеет почти квадратную форму, а другая — очень узкая и длинная, то после отрезания каждого прямоугольника, первая карта будет переворачиваться. Поэтому нужно сделать проверку, что наступил такой случай. Заметим, что в таком случае первая карта действительно будет переворачиваться после каждого отрезания. Поэтому, уменьшение длины первой стороны и второй стороны этой карты будут чередоваться. Этот случай можно обработать бинпоиском, в котором найти максимальное количество отрезаний прямоугольника, что длинная сторона второй карты все еще длиннее короткой. Чтобы найти длину, на которую уменьшится длинная сторона второй карты, нужно найти сумму двух арифметических прогрессий. Первая — сумма длин одной стороны первой карты, вторая — второй.