

Задача А. Задача со звездочкой

Так как ширина обеих ленточек одинакова, достаточно очевидно, что ленточка с наибольшей длиной и будет иметь наибольшую площадь. Давайте найдём максимум из k и $l - k$ и умножим его на ширину S/l , тем самым получим искомую площадь.

Задача В. Творческие задания

Сначала для каждого ученика посчитаем, какое количество пятёрок, полученных за творческие задания, ему нужно для увеличения оценки на 1 балл, обозначив эту величину через X . Давайте найдём минимальную среднюю оценку, которой хватит для поднятия итоговой оценки. Например, если сейчас средняя оценка ученика равна 3.6, то искомое число – это 4.5. Обозначим это число как A . Также посчитаем сумму оценок для каждого ученика, обозначив её как S , и посчитаем количество этих оценок, обозначив это как N .

Тогда справедливо следующей неравенство: $\frac{S+5*X}{N+X} \geq A$. Из него получим, что $X \geq \frac{A*N-S}{5-A}$. X – это целое число, а мы ищем минимальное, поэтому нужно округлить наши вычисления вверх до целого числа.

Теперь осталось лишь отсортировать всех учеников по тому, сколько им нужно получить творческих задач для увеличения своей оценки. Добавлять пятёрки будем тем, кому нужно выдать меньше всего заданий.

Для каждого ученика получаем его новую оценку и находим новую среднюю оценку для класса.

Задача С. Петя и необычный массив

Можно перебрать a_i и потом сделать одно из двух действий:

- Перебрать a_j и выдать a_i и a_j в качестве ответа, если для них выполняются условия $A + B = a_i + a_j$ и $A^B = a_i^{a_j}$. К сожалению, любое правильно написанное решение на Python с таким подходом не пройдет ограничения по времени, но для успешной сдачи задачи с помощью Python можно было воспользоваться подходом ниже.
- Выразить a_j через равенство сумм: $A + B = a_i + a_j \implies a_j = A + B - a_i$. Теперь знаем, какое должно быть a_j при текущем a_i . Нужно проверить, что оно есть в массиве и при этом $i \neq j$. Нужное значение можно найти пробегом по массиву, и тогда получится решение, которое по скорости работает как предыдущее. Также можно заранее подсчитать для каждого элемента массива количество его вхождений (сколько раз элемент входит в массив). Если элементов a_j в списке больше нуля (или единицы, если $a_i = a_j$) и выполняется условие $A^B = a_i^{a_j}$, то можно выводить a_i и a_j в качестве ответа. Условие $A + B = a_i + a_j$ можно не проверять, так как из него выводили a_j .

Примечание: использование собственной реализации небинарного возведения в степень может замедлить ваше решение, однако возведение можно остановить, когда текущее посчитанное значение начнет превышать 10^{12} , так как по условию $A^B \leq 10^{12}$. Тогда подсчет степеней будет менее затратным по времени.

Задача D. Пароль

Давайте поймем, что неограниченное число преобразований позволяет нам превратить изначальную строку в любую другую, поэтому мы можем просто выводить какой-либо подходящий пароль на каждый из тестов. Так как гарантируется, что изначальная строка S не подходит по критериям в качестве пароля, мы можем даже не проверять, что выводимый нами пароль совпадает с S .

Задача E. Забег

Для решения сначала нужно найти периметр круга, используя формулу $2\pi r$. Затем нужно понять, в какой точке от начала на круге сейчас находится Саша. Для этого вычислим остаток от деления x на периметр круга, обозначим его как l . Нам осталось понять, куда смотрит Саша. Обозначим через P периметр круга, тогда:

- если $\frac{P}{8} < l < 3\frac{P}{8}$, то ответ – West

- если $3\frac{P}{8} < l < 5\frac{P}{8}$, то ответ – North
- если $5\frac{P}{8} < l < 7\frac{P}{8}$, то ответ – East
- иначе ответ – South

Задача F. Кроссворд

Для решения данной задачи не нужно перебирать все решения. Входные данные гарантируют, что решение линии существует, поэтому давайте заменим решётку на крестик и проверим, до сих пор ли существует хотя бы одно решение. Если решение до сих пор существует, то мы нашли решение, которое не включает в себя изначальную решётку, и ответ «No». В противном случае мы исключили клетку, которая закрашена во всех возможных решениях линии, и ответ «Yes».

Задача G. Белоснежка и 1000 гномов

Из условия понятно, что скорость движения каждого гнома будет складываться из его собственной скорости и абсолютной угловой скорости гнома под ним. Посчитаем для каждого гнома его абсолютную угловую скорость. Для каждого i -го гнома абсолютная скорость будет равна сумме собственных скоростей от первого гнома до i -го.

Давайте для каждого вопроса будем считать количество умилений с нулевого момента времени до l , оно вычисляется по формуле $l * v_i \text{ div } 360$. Для того, чтобы посчитать до l не включительно, мы должны вычесть единицу из результата в том случае, если $l * v_i$ нацело делится на 360. Теперь то же самое посчитаем для r . Ответом на каждый вопрос будет разность из количества умилений до r и количества умилений до l . Асимптотика данного решения — $O(n + nm)$.

Задача H. 3020

Давайте научимся переводить дату в минуты, прошедшие с начала года. Посмотрим, что и как влияет на ответ.

- Количество минут добавляет в ответ MM минут.
- Каждый час добавляет в ответ 60 минут. Чтобы не путать 12 часов и 00, возьмём остаток от деления количества часов на 12. Если текущее время «До полудня», то количество часов равно $HH \bmod 12$. Если же время «После полудня», то к количеству часов прибавляется 12.
- Каждый день прибавляет к ответу 24 часа или $(dd-1) * 24 * 60$ минут.
- Каждый месяц прибавляет к ответу 30 дней или $(mm-1) * 30 * 24 * 60$ минут.

Давайте переведём все даты в минуты, прошедшие с начала года. Теперь мы можем отсортировать по данному значению любым удобным для нас способом и вывести ответ.

Задача I. Коронавирус

Очевидное решение: на каждом шаге обновляем строку по указанным в условии правилам и делаем это до тех пор, пока строка не будет полностью состоять из 1. К сожалению, такое решение будет очень долгим и не пройдет по времени.

На самом деле ответом будет наибольшее из трёх чисел:

1. Расстояние между началом строки и первой встреченной 1
2. Расстояние между последней встреченной 1 и концом строки
3. Наибольшее расстояние между двумя 1 в строке, деленное на 2 и округленное до целого числа

Таким образом, достаточно один раз пройти по строке, посчитать эти три величины и найти наибольшее из них.