

VIII открытый личный online турнир по
программированию
Юго-Восточного образовательного округа
Центр цифрового образования «ИТ-куб» на базе КОГОАУ
«Вятский многопрофильный лицей» г. Вятские Поляны
3 апреля 2025 года

Разбор задач

Корни уравнения

Количество возможных корней уравнения равно половине исходного числа, округленному вниз.

Таблица умножения

Каждое значение – произведение номера строки и столбца. Достаточно перебрать строки и проверить, есть ли значение $x//i$ в текущей строке i . Необходимо также проверить, не превышает ли это значение имеющееся количество строк, то есть $x//i \leq n$.

Кролики из шляпы

Максимальное количество кроликов, цвет которых удастся угадать, будет равно максимальному элементу в массиве.

Красим по квадратам

По формулировке задачи понятно, что она нахождение НОД, причем классическим способом - вычитанием. Но этот алгоритм при существующих ограничениях, не пройдет по времени. Поэтому применяем алгоритм Евклида - делением.

Параллельно с заменой большего числа на остаток от деления добавляем к результату целую часть от деления, это то количество квадратов, которое мы подсчитываем. Оно и будет в итоге равно количеству разных цветов.

Система оценивания

Можно завести словарь с буквами и их порядковым номером. Не забываем, что начальные буквы алфавита имеют больший вес.

Затем считываем строку, суммируем вес каждой буквы, обращаясь к словарю, делим набранную сумму на длину исходной строки. Помимо этого ищем букву с минимальным весом.

Найденное среднее арифметическое сравниваем с найденным минимальным весом буквы в строке + 1. Если оно будет больше, то числовым результатом будет минимальный вес буквы, увеличенный на 1.

Осталось по полученному числовому результату найти саму букву.

Два массива

В данной задаче можно применить множества. Из объединения множеств вычесть их пересечение и найти длину.

Жадная ворона

Задачу можно решить, используя два указателя. Список необходимо отсортировать. Затем, пока левая и правая границы не сомкнулись, находим сумму левого и правого элементов, и если эта сумма не превышает допустимую, ищем максимум, увеличиваем левую границу. В противном случае – уменьшаем правую границу.

Кар-кар

Последовательно просматриваем список, в котором на каждом шаге находим текущую сумму элементов и текущий максимальный элемент. Только он может быть равен сумме всех других элементов от 1 до i -го.

Счетчик вороньих карканий увеличиваем, если разность между текущей суммой и текущим максимумом равна самому максимуму или текущая сумма равна удвоенному текущему максимуму.

Верните числа

Если попробовать попереставлять числа, меняя на каждом шаге симметричные элементы относительно центра, можно заметить, что по завершению всех операций, элементы, стоящие на четных позициях, останутся на месте. Они будут поменяны четное количество раз. А элементы, стоящие на нечетных местах, будут поменяны с симметричными элементами.

Таким образом, можно за один проход по массиву элементы на четных местах оставить неизменными, а элементы на нечетных позициях, поменять на симметричные.

Счастливым номер

При увеличении числа на 9, 99, 999 и т.д. несколько раз подряд цифра 7 появится как максимум на 7 шаге в разрядах единиц, десятках, сотнях и т.д. То есть, ответом в задаче будет число ≤ 7 .

Можно смоделировать ситуацию добавления к исходному числу числа 9 несколько раз, продолжая до тех пор, пока в числе не появится цифра 7. То же самое делаем с добавлением 99 и т.д. Среди всех операций ищем минимальное значение.

Оно и будет ответом.

Строгая перевозка

На каждом шаге проверяем, войдет ли очередной контейнер на баржу. Если да, то добавляем его к сумме масс контейнеров, отправляемых на этой барже, иначе – баржу отправляем, добавляем к счетчику количества барж единицу, а в счетчик сумм масс контейнеров, претендующих на новую баржу, заносим новую массу контейнера.

Вечер встречи выпускников

Числа необходимо отсортировать.

Для решения можно применить жадный алгоритм. На каждом шаге выбирается лучший вариант.

Понятно, что для минимального итогового количества фотографий лучше, когда все три человека войдут на фото, если не получается, то пробуем разместить двоих, в противном случае берем одного. В зависимости от количества взятых на фото человек, увеличиваем счетчик для просмотра на эту же величину.

Можно задачу решить и динамическим программированием.