Время выполнения задания – 180 минут. Максимальное количество баллов – 100

Задание 1. (25 баллов) Петя очень любит нечетные числа. Родители знают об этом увлечении и поэтому подарили ему набор кубиков. Первым делом Петя убедился, что кубиков нечетное количество N. Он заметил, что на каждом кубике написано по одному нечетному числу от 1 до $2N - 1$, причём числа не повторяются. Шли годы, Петя рос. Когда ему исполнилось 17 лет, он случайно нашёл забытый в шкафу подарок родителей. Время не пощадило когда-то любимую игрушку - потерялись все кубики, на которых были написаны нечетные числа от 1 до $2K - 1$, где K- некоторое число, меньшее N.
Оказавшись в столь непростой ситуации, Петя заинтересовался, является ли сумма чисел на оставшихся кубиках простым числом.

а) Существуют ли числа N и K такие, что сумма чисел на оставшихся кубиках - простое число?

Ответ: Да, например при $N = 3, K = 2$ единственный оставшийся кубик будет иметь на себе цифру 5 - простое число.

б) Существуют ли такие *нечетные* числа N и K, что сумма чисел на оставшихся кубиках - простое число?

Ответ: Заметим, что при N, Kнечетных у нас останется четное количество кубиков (не меньше 2). Очевидно, что сумма при этом будет четна, а значит уже делится на 2. Для того, чтобы сумма была простым числом достаточно доказать, что она будет не меньше 2 (ведь это единственное четное простое число). Заметим, что сумма двух последовательных нечетных чисел уже не меньше 2, ведь
(2a - 1) + (2a + 1) = 4a, где a - натуральное число.

Задание 2. (25 баллов) Володя загадал массив из 100 чисел. Известно, что все числа кроме двух, попарно различны. Алексей хочет найти номера двух совпадающих чисел. Для этого он может спросить у Володи xor любого подотрезка этого массива с длинной строго больше 1, а Володя его сообщит. Подотрезком считается любой набор подряд идущих чисел в массиве. Есть одна проблема, Алексей помнит результат только двух последних вопросов, заданных Володе. Помогите Алексею найти позиции двух совпадающих чисел в этом массиве, задав Володе не более 100000 вопросов.
Подотрезок в вопросе Алексея задается номерами элементов массива.
Напоминаем, что xor двух чисел вычисляется следующим образом: оба числа переводятся в двоичную систему, в результирующем числе на i-ой позиции в двоичной записи стоит 1 тогда и только тогда, когда значения i-ых битов операндов (то есть тех чисел, xor которых мы считаем) различаются. Например$7_{10}\ xor\ 5_{10} = 111_2\ xor\ 10_2 = 101_2 = 5_{10}$
xor подотрезка считается следующим образом: вычисляется xor первых двух чисел, затем считается *xor* результата и третьего числа подотрезка и т.д.
Так xor подотрезка [1, 2, 3, 4] вычисляется следующим образом: $1\ xor\ 2\ xor\ 3\ xor\ 4 = 1_2 xor\ 10_2\ xor\ 11_2\ xor\ 100_2 = 11_2\ xor\ 11_2\ xor\ 100_2 = 0_2\ xor\ 100_2 = 100_2$

Ответ: Обозначим за a искомый массив, а за $a[i]$— i-ый элемент данного массива, также обозначим xor всех элементов с i по j за $xor(i, j)$. Спросим про все подотрезки массива длиной больше трех. Допустим в данный момент мы спрашиваем про подотрезок массива с границами l и r. Для этого подотрезка спросим про подотрезок с границами $l + 1$ и $r - 1$. Заметим, что xor элементов на первом и втором подотрезках совпадают тогда и только тогда, когда $a[l] = a[r]$. Действительно, если $a[l] = a[r]$, то $a[l]\ xor\ a[r] = 0$(вытекает из определения xor), пусть $xor(l + 1, r - 1)$равняется X, тогда $xor(l, r)$ равняется $a[l]\ xor\ X\ xor\ a[r] = (a[l]\ xor\ a[r])\ xor\ X = 0\ xor\ X = X$ (так как xor— коммутативная и ассоциативная операция с нейтральным элементом 0). Теперь пусть $xor(l, r) = xor(l + 1, r - 1) = X$, рассмотрим $0 = (xor(l, r))\ xor\ (xor(l + 1,\ r - 1)) = a[l]\ xor\ a[r]$, первое равенство выполняется, потому что $(xor(l, r))\ xor\ (xor(l + 1,\ r - 1)) = X\ xor\ X = 0$ (из определения xor), второе равенство выполняется так как все элементы из меньшего подотрезка встречаются в большем, а значит их xor обращается в 0 (опять же из-за коммутативности и ассоциативности), значит в результате этого xor останутся только элементы у которых нет пары, а именно $a[l]$ и $a[r]$.Отсюда получаем $a[l]\ xor\ a[r]\ =\ 0$, а из этого уже по определению xor вытекает, что $a[l]\ =\ a[r]$. Таким образом утверждение доказано и как только мы находим пару таких отрезков, то мы находим и равные элементы. Однако данное решение не работает, если равные элементы находятся на расстоянии 1 или 2 ($r\ -\ l\ = 1$ или $r\ -\ l\ =\ 2$), так как в этом случае нужно спрашивать xor пустого подотрезка или подотрезка длины 1, что запрещено условием. Разберем случай, когда $r\ -\ l\ = 1$, для этого спросим xor всех подотрезков с l по $l\ + 1$ (для всех l для которых существует $l\ + 1$элемент), заметим, что если это значение обращается в 0, то мы нашли искомую пару (по доказанному выше). Теперь разберем случай с $r\ -\ l\ =\ 2$, для этого спросим $xor(l, l\ + 1)$ и $xor(l + 1, l\ + 2)$ для всех таких l для которых существует элемент с номером $l\ +\ 2$. Рассмотрим $(xor(l, l\ + 1))\ xor\ (xor(l\ + 1, l\ + 2))\ =\ a[l]\ xor\ a[l\ +\ 1]\ xor\ a[l\ +\ 1]\ xor\ a[l\ +\ 2]\ = a[l]\ xor\ a[l + 2]$,отсюда мы получаем, что если $a[l]\ xor\ a[l + 2]\ =\ 0$, то $a[l]\ =\ a[l\ +\ 2]$. Описанным выше способом для любого расстояния между равными элементами мы научились находить их. Посчитаем количество запросов, которое придется задать. Заметим, что мы рассмотрим все подотрезки массива, кроме подотрезков длины 1, при этом про некоторые из них мы спросим дважды (про некоторые спросим 1 раз, но это не уменьшает оценку). Обозначим за Y количество рассмотренных отрезков, тогда мы сделаем не больше $2\ \cdot Y$запросов. $Y\ =\ C_{100}^{2}$, так как нам нужно выбрать левую границу подотрезка, правую границу подотрезка и порядок нам не важен. $2\ \cdot Y\ = 2\ \cdot \frac{100!}{2!\cdot 98!} =\ 99\ *\ 100\ <\ 100000$. Упражнение: можно ли решить данную задачу за Y запросов?

Задание 3. (25 баллов) У Лёни есть строка длины 50, состоящая из 0 и 1. Он называет строки длины 49, которые состоят из 0 и 2 *шаблонами.* Лёня любит прикладывать шаблоны к строке так, чтобы шаблон не выходил за границы строки. Особенно ему нравится, если каждая двойка в шаблоне покрывает единицу в строке. Такие покрытия он называет *хорошими.* Например, для строки 11100 и шаблона 2000 существует два хороших покрытия: при приложении шаблона, начиная с первого и второго символа.

Сегодня Лёня нашёл очередной шаблон и уже посчитал, сколько существует хороших покрытий с этим шаблоном и своей строкой.

Но Лёня хочет большего: ему интересно, чему равна сумма хороших вхождений по всем парам строк длины 50 и шаблонов длины 49. И вам предлагается помочь Лёне с этим непростым вопросом.

Например, при длине строк равной 2 у Лёни есть 4 различных строки (00, 01, 10 и 11) и 2 различных шаблона (0 и 2). Шаблон 0 не содержит двоек, поэтому его можно приложить в любой из позиций к любой из строк (8 вариантов). Шаблон 2 можно приложить лишь к позициям, на которых стоят единицы (4 варианта). Всего получаем 12 вариантов.

Ответ: Пусть у Лёни есть строка длины Nи шаблон длины K, где $N \geq K$. Рассмотрим произвольный шаблон, в котором ровно n двоек и будем прикладывать его к началу строки. Посчитаем, сколько существует строк для которых такое наложение будет *хорошим.* Под каждой из двоек в строке должны стоять единицы, а во всех остальных позициях может стоять как 0, так и 1. Итого, получаем что для n позиций строки у нас 1 вариант символа, а для остальных $N - n$ есть 2 варианта. Значит таких строк - $2^{N\ -\ n}$. Заметим, что наши рассуждения остаются верными и для приложения шаблона в произвольной из $N - K + 1$позиций.

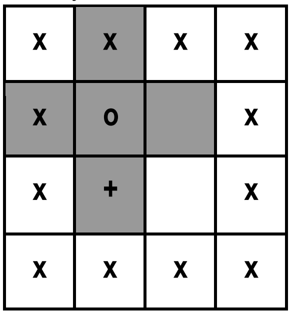
Также мы знаем, что количество шаблонов с n двойками равно $C_{K}^{n} = \frac{K!}{n!(K-n)!}$, то есть количеству способов выбрать подмножество из n позиций для двоек из множества всех позиций (размера K). Теперь соберем всё вместе и преобразуем выражение.

$Answer = (N - K + 1)\sum_{n=0}^{K} C_K^n * 2^{N-n} = (N - K + 1) * 2^{N-K}\sum_{n=0}^{K} C_K^n * 2^{K-n} * 1^n$ =[по биному Ньютона] = $(N - K + 1) * 2^{N-K} * (1 + 2)^K = (N - K + 1) * 2^{N-K} * 3^K$.

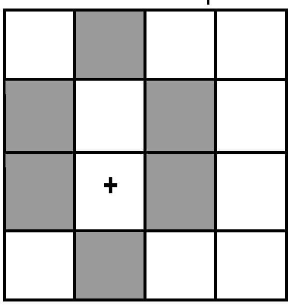
Если подставить числа из задачи, то получим ответ $2 * 2 * 3^{49} = 4 * 3^{49}$.

Задание 4. (25 баллов) Вася начинающий художник, пока он рисует только с помощью простого карандаша и ластика, а в качестве холста использует клетчатый листок бумаги. Вася рисует свои картины следующим образом, сначала он выбирает любую клетку, которая не является крайней на листке, и закрашивает ее карандашом, если она белая, либо стирает карандаш, меняя ее цвет на белый, если она закрашена. Аналогичные алгоритм он проделывает со всеми соседями данной клетки (такими клетками, что имеют общую сторону с выбранными). Он может проделывать эти действия сколько угодно раз, пока ему не надоест. Вася хочет посчитать сколько различных картин он может нарисовать на клетчатом листе размером 64х64 клетки. Вася считает две картины различными, если существует такая клетка, что в одной картине она закрашена, а в другой нет.

Вася не может выбрать ни одну из тех клеток, что отмечены **x**, если он первым шагом выберет клетку, отмеченную **o**, то картина будет выглядеть следующим образом.



Если после этого он выберет клетку, отмеченную **+**, то картина будет выглядеть следующим образом:



После этого Вася может продолжать выбирать любую разрешенную клетку (при желании он может выбирать клетки повторно, например, на третьем шаге выбрать ту клетку, что была отмечена **o**), пока считает это нужным.

Ответ: Будем называть процесс закрашивания клетки и её соседей *операцией*. Рассмотрим какую-то последовательность операций, которые сделал Вася.

Заметим, что цвет каждой клетки однозначно восстанавливается из количества операций, которые имели эффект на неё. Из этого можно сделать вывод, что нам не важно, в каком порядке мы производили операции, нам важно только лишь количество каждой из уникальных операций. Теперь заметим, что применение одной и той же операции Nраз имеет тот же эффект, что и её применение $N mod$ 2раз. Действительно, четность количества перекрашиваний клеток от этого не изменится, а значит не изменится и их цвет. Таким образом, мы можем рассматривать лишь подмножество уникальных операций (это не уменьшит количество итоговых конфигураций). Теперь покажем, что каждое подмножество уникальных операций порождает уникальный рисунок. Предположим, что это не так, то есть мы нашли два различных подмножества уникальных операций, которые сгенерировали одинаковый рисунок.

Рассмотрим все такие операции, которые встречаются только в одном из подмножеств. Из всех таких операций нас интересует самая левая и из самых левых самая верхняя операция (операции будем сравнивать по их центральной клетке).

Пусть центральная клетка этой операции имеет координаты (x, y)(строка и столбец, будем нумеровать их из левого верхнего угла). Рассмотрим клетку $(x, y-1)$. Заметим, что её четность отличается в этих двух подмножествах. Действительно, любая из операций, отличная от операции на (x, y), которая может иметь эффект на $(x, y-1)$, одновременно присутствует или одновременно отсутствует в обоих подмножествах, так как существование операции в одном из подмножеств приводит к противоречию с тем, что левее клетки(x, y)не существует отличающихся операций.

Мы предполагали, что рисунок получился одинаковым, но сейчас мы нашли клетку, цвет которой отличается для этих подмножеств операций. Противоречие. Значит, наше предположение неверно и каждое подмножество уникальных операций приводит нас к уникальному рисунку. Осталось подсчитать количество таких подмножеств. Всего у нас 62^2допустимых операций. Каждую из операций мы можем включить или не включить в подмножество. Таким образом мы получаем ответ: 2^{62^2}.