

Задача А. Двоичный Гаусс

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n битовых векторов размера n и еще один вектор. Получить этот вектор как хог исходных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 300$). В следующих n строках записаны исходные вектора, в последней строке записан вектор, который нужно получить.

Формат выходных данных

Если решений нет, выведите **No solution**. Если решений несколько, выведите **Multiple solutions**. Если решение единственное, выведите номера векторов (вектора нумеруются с 0), которые нужно сложить, чтобы получить данный вектор. Номера выведите в порядке возрастания.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------------|--------------------|
| 3 100 111 101 010 | 1 2 |
| 3 100 111 011 010 | No solution |
| 3 111 010 101 000 | Multiple solutions |

Задача В. Число возможных векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из t битовых векторов размера n ($1 \leq n, t \leq 50$). Сколько различных векторов можно получить как хог этих векторов.

Формат входных данных

В первой строке записаны числа n и t . В следующих t строках записаны вектора.

Формат выходных данных

Выведите число векторов, которые можно получить

Примеры

| стандартный ввод | стандартный вывод |
|---------------------------------|-------------------|
| 2 2 10 11 | 4 |
| 3 4 100 111 011 010 | 8 |
| 3 4 111 111 111 000 | 2 |

Задача С. Добавление векторов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В изначально пустое множество один за другим добавляются m битовых векторов размера n . После добавления каждого скажите, можно ли его представить как хог векторов, добавленных до него.

Формат входных данных

В первой строке записаны числа n и m ($1 \leq n \leq 50$, $1 \leq m \leq 10000$). В следующих m строках записаны вектора.

Формат выходных данных

Для каждого вектора выведите 1, если его можно представить как хог предыдущих и 0 если нельзя.

Примеры

| стандартный ввод | стандартный вывод |
|---------------------------------|-------------------|
| 2 2 10 11 | 0 0 |
| 3 4 100 111 011 010 | 0 0 1 0 |
| 3 4 000 111 111 111 | 1 0 1 1 |

Задача D. Максимальный хог

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан набор из n чисел. Найдите максимальное число, которое можно получить как битовый хог данных.

Формат входных данных

В первой строке записано число n ($1 \leq n \leq 1000$). В следующих n строках записаны исходные числа a_i ($0 \leq a_i \leq 2^{63} - 1$).

Формат выходных данных

Выведите максимальное число, которые можно получить.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------|-------------------|
| 2 1 2 | 3 |
| 3 6 2 8 | 14 |
| 3 15 3 8 | 15 |

Задача E. Полные квадраты

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество A . Определите, сколько непустых подмножеств B множества A являются полноквадратными.

Формат входных данных

В первой строке записано целое число N — количество элементов множества ($1 \leq N \leq 100$). Во второй строке записаны N попарно различных чисел a_i — элементы множества ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

Пример

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 4 49 20 500 7 | 3 |

Задача F. Обобщенные числа фибоначчи

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Мы чуть-чуть обобщили для вас последовательность Фибоначчи, теперь:

$$f_1 = f_2 = 1$$

$$f_i = a \cdot f_{i-1} + b \cdot f_{i-2} + c \cdot 2^i + d \cdot i + e, \text{ для } i > 2$$

Дано n , найдите значение f_n , взятое по модулю $10^9 + 7$.

Формат входных данных

Неотрицательные целые числа: a, b, c, d, e, n . ($0 \leq a, b, c, d, e \leq 10^9$; $1 \leq n \leq 10^{18}$)

Формат выходных данных

Выведите f_n , взятое по модулю $10^9 + 7$.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 1 1 0 0 0 8 | 21 |
| 1 2 3 4 5 6 | 775 |

Задача G. Шоппинг

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 5 секунд |
| Ограничение по памяти: | 256 мегабайт |

Сегодня долгожданный для всех школьников — первый день каникул нового учебного года. Наша главная героиня — Дени — учится в 10 классе. Она хорошо подготовилась к сегодняшнему дню и выяснила, что в центре города находятся N магазинов. Теперь Дени планирует вместе со своими друзьями посетить некоторые из них. В городе есть M пар магазинов (x_i, y_i) , соединенных двусторонними дорогами. Для каждой дороги известно время, которое требуется для перемещения по ней, оно одно и то же для перемещения в обоих направлениях. Никакой магазин не соединен дорогой сам с собой, никакая пара магазинов не соединена более чем одной дорогой.

Дени очень суеверна и одно из её суеверий заключается в том, что она верит, что время, потраченное на перемещения между магазинами, должно нацело делиться на D . При этом Дени с друзьями не может перемещаться между магазинами слишком долго, её путь должен занимать суммарно не больше K . Дени очень любопытна. Она хочет выяснить, сколько существует различных способов начать свой путь в некотором магазине, перемещаться по дорогам между магазинами, и закончить путь в некотором магазине (возможно посещая по пути некоторые магазины и/или дороги более одного раза). Дени помнит, что у нее есть друг-программист — вы — и она просит написать программу, которая вычислит количество корректных способов перемещаться между магазинами. Дени считает способ корректным, если её время в пути не превышает K и делится на D . Вы немедленно указываете Дени, что количество путей может быть слишком большим, поэтому Дени просит вывести остаток от деления количества путей на число 1 000 000 007.

Формат входных данных

На первой строке ввода находятся четыре целых числа N , M , D и K ($2 \leq N \leq 30$, $2 \leq M \leq 435$, $2 \leq D \leq K \leq 10^9$).

На каждой из следующих строк находятся по три целых числа x_i , y_i и t_i ($1 \leq x_i, y_i \leq N$, $1 \leq t_i \leq 10$) — они задают двустороннюю дорогу между магазинами x_i и y_i , перемещение по которой занимает t_i ($1 \leq i \leq M$).

Формат выходных данных

Выведите остаток от деления количества искомых путей на 1 000 000 007.

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 3 3 2 2 1 2 1 2 3 2 3 1 1 | 8 |
| 5 7 5 10 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4 | 58 |
| 5 9 2 20 1 2 1 2 3 2 3 1 1 3 4 1 4 5 2 5 3 1 1 5 1 2 4 1 2 5 1 | 989802661 |
| 5 7 5000000 5000000 1 3 8 2 5 7 3 4 3 1 4 2 2 3 1 1 5 4 4 5 4 | 598634781 |

Задача N. Сумма сумм

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дан массив из n чисел $A_1 \dots A_n$. Вы должны покрасить числа в нём в два цвета: красный и синий. При этом обязательно должно быть хотя бы одно число каждого цвета. После этого *ценностью цвета* называется побитовое исключающее или (XOR) всех чисел этого цвета. А *ценностью массива* называется сумма ценностей синего и красного цвета.

Найдите максимальную возможную ценность массива при оптимальной покраске чисел.

Формат входных данных

В первой строке вводится одно число n ($2 \leq n \leq 10^5$). Во второй строке вводятся n чисел $A_1 \dots A_n$ ($0 \leq A_i < 2^{60}$).

Формат выходных данных

Выведите одно число — максимальную возможную ценность массива.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 3 3 6 5 | 12 |
| 4 23 36 66 65 | 188 |

Замечание

В первом примере нужно покрасить в красный цвет числа 3 и 5, а в синий число 6. Тогда ценности цветов $3 \oplus 5 = 6$ и 6, соответственно.

Задача I. Хорошие раскраски — deluxe edition

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Эта задача не относится к серии задач «Хорошие раскраски», но иных названий для задач я придумать не умею.

Дан гиперграф из n вершин и m ребер. В этой задаче ребра гиперграфа соединяют тройки вершин, i -е ребро соединяет тройку вершин (a_i, b_i, c_i) . Раскраска вершин этого гиперграфа в 3 цвета называется *хорошей*, если для каждого ребра (a, b, c) , выполняется одно из двух условий:

- Все вершины a, b, c раскрашены в один и тот же цвет.
- Все вершины a, b, c раскрашены в попарно различные цвета.

Некоторые вершины гиперграфа уже раскрашены, а именно i -я вершина раскрашена в цвет x_i , если $x_i \neq 0$, и не раскрашена ни в какой цвет, если $x_i = 0$.

Вам требуется определить, существует ли какая-нибудь хорошая раскраска графа, и если да, то вывести любую подходящую.

Формат входных данных

В первой строке указана пара чисел n и m ($1 \leq n \leq 300$, $1 \leq m \leq 2000$) — количество вершин и ребер в гиперграфе.

В последующих m строках указаны тройки чисел a_i, b_i, c_i ($1 \leq a_i < b_i < c_i \leq n$).

В последней строке указано n чисел x_1, \dots, x_n ($0 \leq x_i \leq 3$).

Формат выходных данных

В первой строке выведите Yes или No, в зависимости от того, существует ли хорошая раскраска графа в 3 цвета.

В случае положительного ответа, в следующей строке выведите числа y_1, \dots, y_n ($1 \leq y_i \leq 3$) — найденную раскраску.

Примеры

| стандартный ввод | стандартный вывод |
|--------------------------------------|--------------------|
| 6 2 1 2 3 4 5 6 1 2 0 0 0 0 | Yes 1 2 3 1 1 1 |
| 4 2 1 2 3 2 3 4 1 0 0 2 | No |

Задача J. Исключаем или сдвигаем

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Множество целых неотрицательных чисел S изначально состоит из n элементов $A_1 \dots A_n$. Можно добавлять числа в множество по следующим операциям:

- Если $B \in S$, то можно добавить $2B$ в S .
- Если $B \in S, C \in S$, то можно добавить $B \oplus C$ в S .

Здесь под \oplus понимается операция XOR (побитовое исключающее или).

Дано число X . Требуется посчитать количество чисел, не больших X , которые могут оказаться в S после какого-то количества операций. Так как это число может быть очень большим, от вас требуется посчитать его по модулю 998244353.

Формат входных данных

В первой строке даны два числа n и X ($1 \leq n \leq 6, 0 \leq X < 2^{4000}$). Число n записано в десятичной записи, а X в двоичной. В следующих n строках даны числа A_i ($0 \leq A_i < 2^{4000}$). Числа A_i тоже записаны в двоичной записи.

Формат выходных данных

Выведите одно число — количество чисел, не больших X , которые могут оказаться в множестве, по модулю 998244353.

Примеры

| стандартный ввод | стандартный вывод |
|---|-------------------|
| 3 111 1111 10111 10010 | 4 |
| 4 100100 1011 1110 110101 1010110 | 37 |
| 4 111001100101001 10111110 1001000110 100000101 11110000011 | 1843 |

Задача К. Кузнечик

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Кузнечик двигается из клетки 0 в клетку n . За один прыжок он может прыгать на 1, 2 или 3 клетки вперед. Ваша задача — посчитать, сколько способов допрыгать у него есть. Ситуация усложняется тем, что иногда некоторые клетки становятся непроходимыми, а иногда наоборот. Вам нужно быстро пересчитывать число способов добраться из 0 в n после каждого изменения.

Формат входных данных

Первая строка содержит два числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) — число клеток и число изменений. Следующие m строк содержат описания изменений. Каждое изменение описывается числом x ($1 \leq x \leq n - 1$), номером клетки, для которой изменился статус (то есть если она была проходимой, то стала непроходимой и наоборот).

Формат выходных данных

Выведите $m + 1$ число — число способов добраться из 0 до n в начале и после каждого изменения по модулю $10^9 + 7$.

Примеры

| стандартный ввод | стандартный вывод |
|-------------------------|------------------------------------|
| 3 4 1 2 1 2 | 4 2 1 2 4 |
| 100 2 33 56 | 347873931 85092847 684577937 |