

Задача А. Сумма функций XOR

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

Дан массив a длины n , состоящий из неотрицательных целых чисел.

Необходимо вычислить значение $\sum_{l=1}^n \sum_{r=l}^n f(l, r) \cdot (r - l + 1)$, где $f(l, r)$ равно $a_l \oplus a_{l+1} \oplus \dots \oplus a_{r-1} \oplus a_r$ (символ $\oplus$ обозначает побитовое исключающее ИЛИ).

Так как ответ может быть очень большим, выведите его по модулю 998244353.

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 3 \cdot 10^5$) — длину массива a .

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — значение $\sum_{l=1}^n \sum_{r=l}^n f(l, r) \cdot (r - l + 1)$, взятое по модулю 998244353.

Примеры

стандартный ввод	стандартный вывод
3 1 3 2	12
4 39 68 31 80	1337
7 313539461 779847196 221612534 488613315	257421502 633203958 394620685 761188160

Замечание

В первом примере ответ равен $f(1, 1) + 2 \cdot f(1, 2) + 3 \cdot f(1, 3) + f(2, 2) + 2 \cdot f(2, 3) + f(3, 3) = 1 + 2 \cdot 2 + 3 \cdot 0 + 3 + 2 \cdot 1 + 2 = 12$.

Задача В. Роблокс

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

В роблоксе появилась новая мини игра. В неё играют 3 игрока, пронумерованные от 1 до 3. У каждого из игроков есть стопка из некоторого количества карт. На каждой карте написано какое-то число от 1 до 3. У первого игрока в стопке a карт, у второго – b , у третьего – c . Игрок не может перемешивать карты в своей стопке.

Игра проходит по следующим правилам:

- Игроки ходят по очереди. Первым ходит игрок под номером 1.
- Если у текущего игрока остались карты в стопке, то он сбрасывает верхнюю из них. Затем игрок, чей номер указан на этой карте начинает свой ход.
- Если у текущего игрока закончились карты, он побеждает.

Существуют 3^{a+b+c} возможных начальных конфигураций колод 3 игроков. Сколько из них приведут игрока с номером 1 к победе.

Поскольку ответ может быть слишком большим, выведите его по модулю $10^9 + 7$.

Формат входных данных

В единственной строке содержатся 3 целых числа a, b, c ($1 \leq a, b, c \leq 3 \cdot 10^5$).

Формат выходных данных

Выведите ответ по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 1 1	17
4 2 2	1227
1000 1000 1000	261790852

Задача С. Ультра-мяу

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 256 мегабайт

Барсик подарил вам массив a длины n , состоящий из чисел $1, 2, \dots, n$. Принять? Конечно, да! Но что с ним делать? Конечно же посчитать $\text{MEOW}(a)$.

Пусть $\text{MEX}(S, k)$ — k -е целое **положительное** число в порядке возрастания, отсутствующее в множестве S . Обозначим за $\text{MEOW}(a)$ сумму $\text{MEX}(b, |b| + 1)$, по всем **различным** подмножествам b массива a .

Примеры значений $\text{MEX}(S, k)$ для множеств:

- $\text{MEX}(\{3, 2\}, 1) = 1$, т. к. 1 — первое целое положительное число, отсутствующее в множестве;
- $\text{MEX}(\{4, 2, 1\}, 2) = 5$, т. к. первые два целые положительные числа, которых нет в множестве, это 3 и 5;
- $\text{MEX}(\{\}, 4) = 4$, т. к. в пустом множестве нет никаких чисел, следовательно первые целые положительные 4 числа, которых в нём нет, это 1, 2, 3, 4.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

В единственной строке каждого набора входных данных вводится целое число n ($1 \leq n \leq 5000$), размер массива подаренных чисел.

Гарантируется, что сумма n^2 по всем наборам входных данных не превосходит $25 \cdot 10^6$

Формат выходных данных

Для каждого набора входных данных выведите одно число — $\text{MEOW}(a)$. Поскольку оно может оказаться очень большим, выведите его по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
5	12
2	31
3	354226409
4999	184
5	4
1	

Задача D. А листья падают...

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

Дано дерево из n вершин, пронумерованных целыми числами от 1 до n . Изначально множество S пустое.

Будем последовательно удалять вершины из дерева. На каждом шаге, пока в дереве находится более одной вершины, выполняются следующие действия:

1. среди всех листьев текущего дерева определяется вершина x с наибольшим номером;
2. вершина x добавляется в множество S ;
3. после этого удаляется произвольный лист текущего дерева, номер которого отличен от x .

Листом считается вершина, степень которой в текущем дереве не превосходит единицы.

Добавление уже содержащегося в S элемента не изменяет множество. После удаления вершины удаляется также инцидентное ей ребро, если оно существует.

Процесс состоит ровно из $n - 1$ шагов, поэтому после его завершения в дереве остаётся одна вершина.

В зависимости от выбора удаляемого листа на каждом шаге итоговое множество S может быть различным.

Требуется определить количество различных множеств S , которые могут быть получены после завершения процесса. Выведите ответ по модулю 998, 244, 353.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество вершин дерева.

В следующих $n - 1$ строках содержатся по два целых числа u и v ($1 \leq u, v \leq n$, $u \neq v$), означающих, что вершины u и v соединены ребром.

Гарантируется, что заданный граф является деревом.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — количество различных итоговых множеств S по модулю 998, 244, 353.

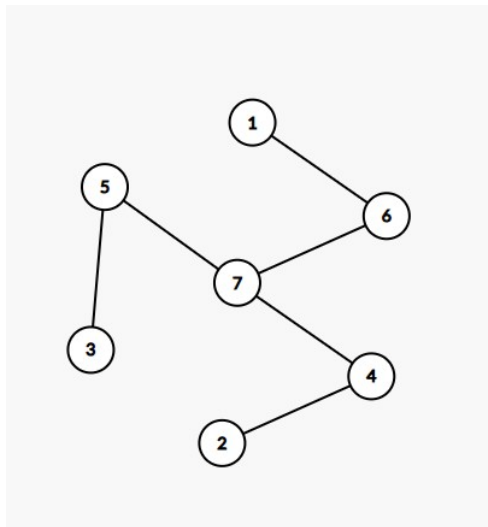
Пример

стандартный ввод	стандартный вывод
6	1
2	1
1 2	3
5	13
5 1	1
5 3	1
5 4	
5 2	
7	
7 6	
7 4	
5 7	
1 6	
2 4	
3 5	
10	
10 9	
10 8	
10 7	
10 6	
9 5	
8 4	
7 3	
6 2	
5 1	
4	
2 4	
3 1	
1 4	
4	
1 4	
2 3	
3 4	

Замечание

В первом наборе входных данных возможен только один вариант удаления: удаляется вершина 1. В этот момент листом с наибольшим номером является вершина 2, поэтому итоговое множество равно 2.

В третьем наборе входных данных дерево имеет следующий вид:



Возможны следующие итоговые множества:

- 3, 6, 7, если удалять вершины в порядке 1, 2, 3, 4, 5, 6;
- 3, 4, 6, 7, если удалять вершины в порядке 2, 1, 3, 4, 5, 6;
- 3, 4, 5, 6, 7, если удалять вершины в порядке 2, 3, 1, 4, 5, 6.

Других итоговых множеств получить нельзя.

Задача E. Magic in my bones

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

Нюша бросила n игральных костей, у каждой из которых k сторон с числами от 1 до k . Игральные кости неразличимы. Два исхода бросков считаются одинаковыми, если для каждого x количество костей, показавших число x , совпадает.

Требуется для каждого i от 2 до $2 \cdot k$ найти количество различных исходов, при которых никакие две игральные кости не дают в сумме число i . То есть не существует пары выпавших чисел (a, b) (a может быть равно b , но они должны выпасть на двух разных игральных костях), таких что $a + b = i$.

Так как ответ может быть очень большим, выведите его по модулю 998244353.

Формат входных данных

В единственной строке даны 2 целых числа k и n ($1 \leq k, \leq 2000, 2 \leq n \leq 2000$).

Формат выходных данных

Выведите $2 \cdot k - 1$ чисел в различных строках. В t -й строке ($1 \leq t \leq 2 \cdot k - 1$) выведите ответ для $i = t + 1$.

Примеры

стандартный ввод	стандартный вывод
3 3	7 7 4 7 7
4 5	36 36 20 20 20 36 36
6 1000	149393349 149393349 668669001 668669001 4000002 4000002 4000002 668669001 668669001 149393349 149393349

Замечание

- Для $i = 2$ возможные варианты: $(1, 2, 2), (1, 2, 3), (1, 3, 3), (2, 2, 2), (2, 2, 3), (2, 3, 3), (3, 3, 3)$. Ответ 7.
- Для $i = 3$ возможные варианты: $(1, 1, 1), (1, 1, 3), (1, 3, 3), (2, 2, 2), (2, 2, 3), (2, 3, 3), (3, 3, 3)$. Ответ 7.

- Для $i = 4$ возможные варианты: $(1, 1, 1)$, $(1, 1, 2)$, $(2, 3, 3)$, $(3, 3, 3)$. Ответ 4.

Задача F. МЕХимизируйте счёт

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

Предположим, что мы разбиваем элементы массива b на k непустых мультимножеств $S_1, S_2, \dots, S_k$, где k — произвольное целое положительное число. Определим *счёт* b как максимальное значение $\text{MEX}(S_1) + \text{MEX}(S_2) + \dots + \text{MEX}(S_k)$ среди всех возможных разбиений b для любого целого числа k .

Вашему завистнику дан массив a длины n . Поскольку он знает, что вычислить *счёт* массива a для вас слишком просто, он просит вас вычислить сумму *счётов* всех $2^n - 1$ непустых подпоследовательностей массива a .[†] Поскольку ответ может быть очень большим, выведите его по модулю 998 244 353.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина a .

Вторая строка каждого набора входных данных содержит n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i < n$) — элементы массива a .

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите ответ по модулю 998 244 353.

Пример

стандартный ввод	стандартный вывод
4	11
3	26
0 0 1	53
4	0
0 0 1 1	
5	
0 0 1 2 2	
4	
1 1 1 1	

Замечание

В первом наборе входных данных мы должны рассмотреть семь подпоследовательностей:

- $[0]$: Счёт равен 1.
- $[0]$: Счёт равен 1.
- $[1]$: Счёт равен 0.
- $[0, 0]$: Счёт равен 2.
- $[0, 1]$: Счёт равен 2.

*МЕХ набора целых чисел $c_1, c_2, \dots, c_k$ определяется как наименьшее неотрицательное целое число x , которое не встречается в наборе c . Например, $\text{MEX}([0, 1, 2, 2]) = 3$ и $\text{MEX}([1, 2, 2]) = 0$

[†]Последовательность x является подпоследовательностью последовательности y , если x можно получить из y , удалив несколько (возможно, ноль или все) элементов.

- $[0, 1]$: Счёт равен 2.
- $[0, 0, 1]$: Счёт равен 3.

Ответ для первого набора входных данных составляет $1 + 1 + 2 + 2 + 2 + 3 = 11$.

В последнем наборе входных данных все подпоследовательности имеют счёт 0.

Задача G. Бинарные строки и блоки

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

Назовем **блоком** в бинарной строке (строке, состоящей из символов 0 и/или 1) ее непрерывную подстроку из символов одного и того же типа, которую нельзя продлить ни влево, ни вправо. Например, в строке 110001111 три блока:

- 11 (с 1-го символа по 2-й символ);
- 000 (с 3-го символа по 5-й символ);
- 1111 (с 6-го символа по 9-й символ).

Подстрока с 7-го символа по 9-й символ 111 не является блоком, потому что ее можно продлить влево. Подстрока с 1-го символа по 5-й символ 11000 не является блоком, потому что она содержит символы разных типов.

Назовем строку *красивой*, если из нее можно удалить **ровно один блок** так, чтобы получилась строка из **нечетного** количества блоков. Например:

- строка 110001111 красивая, потому что можно удалить блок с 3-го по 5-й символ, и получится строка 111111, состоящая из одного блока;
- строка 1010 красивая, потому что можно удалить блок с 1-го по 1-й символ, и получится строка 010, состоящая из трех блоков;
- строка 0000 некрасивая, потому что единственный способ удалить из нее блок приведет к тому, что мы получим пустую строку, а она состоит из 0 блоков.

Дано число n и m ограничений, i -е из которых описывается парой целых чисел l_i, r_i . Обозначим за $s[l : r]$ подстроку строки s с символа l по символ r включительно, то есть $s[l : r] = s_l s_{l+1} \dots s_r$. Ваша задача — посчитать количество бинарных строк s длины n , удовлетворяющих следующему условию:

- для каждого i от 1 до m подстрока $s[l_i : r_i]$ является красивой.

Формат входных данных

В первой строке задано одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

В первой строке каждого набора входных данных заданы два целых числа n и m ($2 \leq n \leq 3 \cdot 10^5$; $1 \leq m \leq 3 \cdot 10^5$) — требуемая длина строки и количество ограничений, соответственно.

Далее следуют m строк, i -я из которых содержит два целых числа l_i, r_i ($1 \leq l_i < r_i \leq n$) — описание i -го ограничения.

Дополнительные ограничения на входные данные:

- сумма n по всем наборам не превосходит $3 \cdot 10^5$;
- сумма m по всем наборам не превосходит $3 \cdot 10^5$.

Формат выходных данных

На каждый набор входных данных выведите одно целое число — количество строк, удовлетворяющих всем ограничениям. Так как оно может быть огромным, выведите его по модулю 998244353.

Пример

стандартный ввод	стандартный вывод
3	2
4 3	4
1 2	570529459
2 3	
3 4	
4 2	
1 2	
3 4	
200 1	
13 37	

Замечание

В первом примере из условия подходят следующие строки: 1010, 0101. Для каждой из этих строк и $s[1 : 2]$, и $s[2 : 3]$, и $s[3 : 4]$ являются красивыми.

Задача Н. Шарики

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Монокарп выложил на стол слева направо n шариков, причем цвет i -го слева шарика равен a_i . Он любит порядок, поэтому захотел, чтобы все шарики одного цвета образовывали непрерывные отрезки, причем для каждого цвета этот отрезок был ровно один.

Иными словами, Монокарп хочет расположить шарики так, что для каждого цвета j , если самый левый шарик цвета j находится в позиции l , а самый правый — в позиции r , то все шарики между позициями l и r должны иметь цвет j . Такое расположение шариков Монокарп называет упорядоченным.

Для упорядочивания шариков Монокарп может выполнять следующую операцию: выбрать пару соседних шариков и поменять их местами.

Перед вами стоит задача определить минимальное количество операций, которые должен сделать Монокарп, чтобы упорядочить шарики на столе. Обратите внимание, что не важно, в каком именно порядке будут расположены отрезки, состоящие из одноцветных шариков, главное, чтобы все шарики одинакового цвета образовывали непрерывные отрезки, и для каждого цвета такой отрезок был ровно один.

Формат входных данных

В первой строке следует целое число n ($2 \leq n \leq 4 \cdot 10^5$) — количество шариков на столе.

Во второй строке следует последовательность $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 20$), где a_i равно цвету i -го шарика.

Формат выходных данных

Выведите минимальное количество обменов соседних шариков, которые должен сделать Монокарп, чтобы упорядочить шарики на столе.

Примеры

стандартный ввод	стандартный вывод
7 3 4 2 3 4 2 2	3
5 20 1 14 10 2	0
13 5 5 4 4 3 5 7 6 5 4 4 6 5	21

Замечание

В первом примере Монокарпу достаточно сделать три обмена. Сначала он может поменять местами третий и четвертый шарик, после этого последовательность шариков станет равна $[3, 4, 3, 2, 4, 2, 2]$. Затем Монокарп может поменять местами второй и третий шарик, после этого последовательность шариков станет равна $[3, 3, 4, 2, 4, 2, 2]$. После этого Монокарпу остается поменять местами четвертый и пятый шарик, после этого последовательность шариков станет равна $[3, 3, 4, 4, 2, 2, 2]$.

Во втором примере ничего менять не нужно, так как все шарик уже лежат в удовлетворяющем Монокарпа порядке.

Задача I. Раскраски дерева

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Рассмотрим корневое неориентированное дерево. Каждую вершину можно раскрасить в синий, зелёный или жёлтый цвет. Раскраска называется *красивой*, если выполняются следующие условия:

- корень дерева окрашен в зелёный цвет;
- если рассмотреть **все синие и зелёные вершины**, то они достижимы друг из друга, не проходя через **жёлтые** вершины;
- если рассмотреть **все жёлтые и зелёные вершины**, то они достижимы друг из друга, не проходя через **синие** вершины;

Вам дано целое число m . Ваша задача — вычислить минимальное количество вершин в дереве, для которого существует **ровно m красивых** раскрасок.

Формат входных данных

Первая строка содержит одно целое число ($1 \leq t \leq 10^5$) — количество наборов входных данных.

В единственной строке каждого набора входных данных содержится одно целое число m ($1 \leq m \leq 5 \cdot 10^5$).

Формат выходных данных

Для каждого набора входных данных выведите одно целое число — минимальное количество вершин в дереве, для которого существует **ровно m красивых** раскрасок. Если такого дерева не существует, выведите -1 .

Пример

стандартный ввод	стандартный вывод
5	1
1	2
3	3
5	4
7	3
9	

Замечание

Пусть g обозначает зелёный цвет, b — синий, а y — жёлтый.

В первом примере рассмотрим простое дерево из 1 вершины. У этого дерева есть ровно 1 красивое раскрашивание: корень зелёный.

Во втором примере рассмотрим простое дерево с 2 вершинами и корнем в 1-й вершине. Существует ровно 3 красивых раскрашивания: $[g, g]$, $[g, b]$ и $[g, y]$.

В третьем примере рассмотрим дерево-цепочку с 3 вершинами и корнем в 1-й вершине. Существует ровно 5 красивых раскрашиваний: $[g, g, g]$, $[g, g, b]$, $[g, g, y]$, $[g, b, b]$ и $[g, y, y]$.

В пятом примере рассмотрим дерево с 3 вершинами и корнем в 1-й вершине, к которому присоединены остальные 2 вершины. Существует ровно 9 красивых раскрашиваний: $[g, g, g]$, $[g, g, b]$, $[g, g, y]$, $[g, b, g]$, $[g, b, b]$, $[g, b, y]$, $[g, y, g]$, $[g, y, b]$ и $[g, y, y]$.