

Е. Минимизация разности

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

В данной задаче вам задана последовательность a_1, a_2, \dots, a_n , состоящая из n целых чисел.

Вы можете изменять элементы последовательности. В ходе одной операции изменения, вы можете выбрать произвольную позицию в последовательности и увеличить или уменьшить элемент, находящийся в этой позиции, на единицу.

Перед вами стоит задача определить минимально возможную разность между максимальным и минимальным элементами последовательности, которую можно достичь, произведя **не более** k описанных операций.

Входные данные

В первой строке следуют два целых числа n и k ($2 \leq n \leq 10^5, 1 \leq k \leq 10^{14}$) — количество элементов в последовательности и максимальное количество операций изменения элементов, которое можно произвести.

Во второй строке следует последовательность целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

Выходные данные

Выведите минимально возможную разность между максимальным и минимальным элементами последовательности, которую можно достичь, произведя **не более** k операций, описанных в условии.

Примеры

входные данные	Скопировать
4 5 3 1 7 5	
выходные данные	Скопировать
2	
входные данные	Скопировать
3 10 100 100 100	
выходные данные	Скопировать
0	
входные данные	Скопировать
10 9 4 5 5 7 5 4 5 2 4 3	
выходные данные	Скопировать
1	

Примечание

В первом примере можно дважды увеличить на единицу второе число и дважды уменьшить на единицу третье число. Тогда получим последовательности $[3, 3, 5, 5]$, в которой разность между максимальным и минимальным элементами равна 2. При этом останется одна еще операция, но ее можно не использовать, так как не удастся сделать разность между максимальным и минимальным элементами меньше 2.

Во втором примере можно не использовать ни одной операции, так как все элементы в последовательности одинаковые, а значит разность между максимальным и минимальным элементами равна 0.



D. 505

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт

Бинарная матрица называется **хорошей**, если каждая квадратная подматрица с **четной** длиной стороны содержит **нечетное** число единиц.

Для данной бинарной матрицы a , состоящей из n строк и m столбцов, определите минимальное количество ячеек, которые нужно изменить, чтобы сделать ее хорошей, или сообщите, что ее вообще нельзя сделать хорошей.

Все вышеприведенные термины имеют свои обычные значения — обратитесь к разделу Примечания для их формальных определений.

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n \leq m \leq 10^6$ и $n \cdot m \leq 10^6$) — число строк и столбцов в a соответственно.

Каждая из следующих n строк содержит m символов, каждый из которых равен 0 или 1. Если j -й символ в i -й строке равен 1, то $a_{i,j} = 1$. Аналогично, если j -й символ в i -й строке равен 0, то $a_{i,j} = 0$

Выходные данные

Выведите минимальное количество ячеек, которое нужно изменить, чтобы сделать a хорошей, или выведите -1 , если это вообще невозможно.

Примеры

входные данные	Скопировать
3 3 101 001 110	
выходные данные	Скопировать
2	

входные данные	Скопировать
7 15 000100001010010 100111010110001 101101111100100 010000111111010 111010010100001 000011001111101 11111011010011	
выходные данные	Скопировать
-1	

Примечание

В первом случае достаточно заменить $a_{1,1}$ на 0 и $a_{2,2}$ на 1.

Вы можете проверить, что нет никакой возможности сделать матрицу во втором случае хорошей.

Некоторые определения —

- Бинарная матрица — это матрица, в которой каждый элемент равен 1 или 0.
- Подматрица описывается 4 параметрами — r_1, r_2, c_1 и c_2 ; здесь $1 \leq r_1 \leq r_2 \leq n$ и $1 \leq c_1 \leq c_2 \leq m$.
- Эта подматрица содержит все элементы $a_{i,j}$, которые удовлетворяют как $r_1 \leq i \leq r_2$ так и $c_1 \leq j \leq c_2$.
- Подматрица называется квадратом четной длины, если $r_2 - r_1 = c_2 - c_1$ и $r_2 - r_1 + 1$ делится на 2.

Е. Увеличение частоты

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам задан массив a длины n . Вы можете выбрать один отрезок $[l, r]$ ($1 \leq l \leq r \leq n$) и целое число k (положительное, отрицательное или ноль) и изменить a_l, a_{l+1}, \dots, a_r на k каждое (другими словами, $a_i := a_i + k$ для каждого $l \leq i \leq r$).

Какое максимально возможное количество элементов со значением c может быть получено после одной такой операции?

Входные данные

Первая строка содержит два целых числа n и c ($1 \leq n \leq 5 \cdot 10^5, 1 \leq c \leq 5 \cdot 10^5$) — длина массив и значение c .

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 5 \cdot 10^5$) — массив a .

Выходные данные

Выведите единственное число — максимально возможное количество элементов со значением c , которое может быть получено после одной операции описанной выше.

Примеры

входные данные	Скопировать
6 9 9 9 9 9 9 9	
выходные данные	Скопировать
6	

входные данные	Скопировать
3 2 6 2 6	
выходные данные	Скопировать
2	

Примечание

В первом примере можно выбрать любой отрезок и $k = 0$. Массив после применения операции не изменится.

Во втором примере можно выбрать отрезок $[1, 3]$ и $k = -4$. Массив примет вид $[2, -2, 2]$.



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Следи за большим средним

ограничение по времени на тест: 1.5 секунд
ограничение по памяти на тест: 256 мегабайт

Вам дан массив целых чисел a_1, a_2, \dots, a_n и целое число x .

Вам нужно выбрать максимальное количество элементов массива таким образом, чтобы для любого подотрезка массива a_l, a_{l+1}, \dots, a_r , который содержит строго больше одного элемента ($l < r$), выполняется хотя бы одно из двух:

- Хотя бы один элемент на этом отрезке **не** выбран, или
- $a_l + a_{l+1} + \dots + a_r \geq x \cdot (r - l + 1)$.

Входные данные

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

Далее следуют описания t наборов, по три строки на набор.

Первая строка содержит одно целое число n ($1 \leq n \leq 50\,000$): размер массива.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($-100\,000 \leq a_i \leq 100\,000$).

Третья строка содержит одно целое число x ($-100\,000 \leq x \leq 100\,000$).

Выходные данные

Для каждого набора входных данных выведите одно целое число: максимальное количество элементов, которое вы можете выбрать.

Пример

входные данные	Скопировать
4 5 1 2 3 4 5 2 10 2 4 2 4 2 4 2 4 2 4 3 3 -10 -5 -10 -8 3 9 9 -3 5	
выходные данные	Скопировать
4 8 2 2	

Примечание

В первом примере один из корректных способов выбрать элементы следующий: $[1, 2, 3, 4, 5]$. Все подотрезки удовлетворяют хотя бы одному критерию. Например, в подотрезке $l = 1, r = 2$ элемент 2 не выбран, то есть выполнен первый критерий. А в подотрезке $l = 3, r = 5$ выполняется $3 + 4 + 5 = 12 \geq 2 \cdot 3$, удовлетворяя второй критерий.

Выбрать все элементы нельзя, так как в этом случае для $l = 1, r = 2$ все элементы будут выбраны, и $a_1 + a_2 = 3 < 2 \cdot 2$. Таким образом, максимальное количество выбранных элементов равно 4.

Во втором примере один из корректных способов выбрать элементы следующий: $[2, 4, 2, 4, 2, 4, 2, 4]$.

В третьем примере один из корректных способов выбрать элементы следующий: $[-10, -5, -10]$.

В четвертом примере один из корректных способов выбрать элементы следующий: $[9, 9, -3]$.

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

D. Для геймеров. От геймеров.

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 256 мегабайт

Монокарп играет в стратегическую игру. В игре он собирает отряд для борьбы с монстрами. Перед каждой битвой у Монокарпа есть C монет, которые он может потратить на свой отряд.

Перед началом каждой битвы в его отряде никого нет. Монокарп выбирает **один тип воинов** и набирает не больше воинов этого типа, чем он может себе позволить за C монет.

Существует n типов воинов. Каждый тип воинов имеет три параметра:

- c_i — стоимость одного воина i -го типа;
- d_i — урон, который наносит один воин i -го типа за секунду;
- h_i — здоровье одного воина i -го типа.

Монокарпу придется сразиться с m монстрами. У каждого монстра есть два параметра:

- D_j — урон, который наносит j -й монстр за секунду;
- H_j — здоровье j -го монстра.

Монокарп сражается только с j -м монстром во время j -й битвы. Он хочет, чтобы все его воины остались в живых. И отряд Монокарпа, и монстр атакуют одновременно и непрерывно. Таким образом, Монокарп выигрывает битву тогда и только тогда, когда его отряд убивает монстра строго быстрее, чем монстр убивает одного из его воинов. Время сравнивается без округления.

Для каждого монстра Монокарп хочет знать минимальное количество монет, которое он должен потратить, чтобы убить этого монстра. Если это количество больше, чем C , то сообщите, что убить этого монстра невозможно.

Входные данные

В первой строке записаны два целых числа n и C ($1 \leq n \leq 3 \cdot 10^5$; $1 \leq C \leq 10^6$) — количество типов воинов и количество монет у Монокарпа перед каждой битвой.

В i -й из следующих n строк записаны три целых числа c_i, d_i и h_i ($1 \leq c_i \leq C$; $1 \leq d_i, h_i \leq 10^6$).

В следующей строке записано одно целое число m ($1 \leq m \leq 3 \cdot 10^5$) — количество монстров, с которыми придется сразиться Монокарпу.

В j -й из следующих m строк записаны два целых числа D_j и H_j ($1 \leq D_j \leq 10^6$; $1 \leq H_j \leq 10^{12}$).

Выходные данные

Выведите m целых чисел. Для каждого монстра выведите минимальное количество монет, которое Монокарп должен потратить, чтобы убить этого монстра. Если это количество больше, чем C , то выведите -1 .

Примеры

входные данные	Скопировать
3 10 3 4 6 5 5 5 10 3 4 3 8 3 5 4 10 15	
выходные данные	Скопировать
5 3 -1	

входные данные	Скопировать
5 15 14 10 3 9 2 2 10 4 3 7 3 5 4 3 1 6	

11 2 1 1 4 7 2 1 1 14 3 3	
выходные данные	Скопировать
14 4 14 4 7 7	

входные данные	Скопировать
5 13 13 1 9 6 4 5 12 18 4 9 13 2 5 4 5 2 16 3 6 2	
выходные данные	Скопировать
12 5	

Примечание

Рассмотрим первого монстра из первого примера.

Монокарп не может взять одного воина первого типа, потому что и ему, и монстру потребуется 0.75 секунды, чтобы убить друг друга. Он может взять двух воинов стоимостью 6 монет и убить монстра за 0.375 секунду.

Монокарп может взять одного воина второго типа, потому что он убивает монстра за 0.6 секунды, а монстр убивает его за 0.625 секунды. Воин быстрее. Таким образом, достаточно 5 монет.

Монокарпу понадобится как минимум три воина третьего типа, чтобы убить первого монстра, что будет стоить 30 монет.

Монокарп потратит меньше всего монет, если выберет второй тип воинов и возьмет одного воина этого типа.

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 26.07.2025 11:31:13^{UTC+5} (i1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



G. Деньги теперь покупают меньше счастья

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Никогда нельзя купить достаточно счастья, поэтому мы снова здесь! В этой версии вы можете купить только $h_i = 1$ единицу счастья каждый месяц, но количество месяцев увеличено в разы. Мы находимся в области квантового счастья и временного растяжения.

Будучи физиком, Чарли любит планировать свою жизнь в простых и точных терминах.

В течение следующих m месяцев, начиная с нулевой суммы денег, Чарли будет усердно работать и зарабатывать x фунтов в месяц. Для i -го месяца ($1 \leq i \leq m$) будет одна возможность заплатить стоимость c_i фунтов за получение одной единицы счастья. Вы не можете купить более одной единицы счастья каждый месяц.

Запрещено брать в долг. Деньги, заработанные в i -м месяце, могут быть потрачены только в более поздний j -й месяц ($j > i$).

Поскольку физики не пишут код, помогите Чарли найти максимально достижимое количество единиц счастья.

Входные данные

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

Первая строка каждого набора содержит два целых числа, m и x ($1 \leq m \leq 2 \cdot 10^5$, $1 \leq x \leq 10^3$) — общее количество месяцев и ежемесячная зарплата.

Вторая строка каждого набора содержит m целых чисел c_1, c_2, \dots, c_m ($1 \leq c_i \leq 10^3$) — стоимость одной единицы счастья для каждого месяца.

Гарантируется, что сумма m по всем наборам входных данных теста не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно целое число — максимальное количество счастья, которое может получить Чарли.

Пример

входные данные	Скопировать
6 3 3 2 2 2 6 5 2 2 8 2 6 8 6 4 4 10 3 8 6 10 2 1 1 1 4 1 4 1 3 1 4 2 1 3 4 3	
выходные данные	Скопировать
2 4 3 1 2 2 1	





[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

Е. Пути и деревья

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

Маленькая девочка Сьюзи случайно нашла тетрадь своего старшего брата. У неё есть много более важных дел, чем решение задач, но эта показалась ей слишком интересной, так что она всё же захотела узнать её решение и решила спросить у вас. Итак, условие задачи.

Пусть дан связный взвешенный неориентированный граф $G = (V, E)$ (здесь V — множество вершин, E — множество рёбер). Деревом кратчайших путей из вершины u называется граф $G_1 = (V, E_1)$, который является деревом, множество его рёбер E_1 является подмножеством множества рёбер исходного графа E , и длины кратчайших путей от u до любой вершины в G и в G_1 совпадают.

Вам дан связный взвешенный неориентированный граф G и вершина u . Необходимо найти дерево кратчайших путей заданного графа из вершины u , суммарный вес рёбер которого минимален.

Входные данные

В первой строке находятся два числа n и m ($1 \leq n \leq 3 \cdot 10^5, 0 \leq m \leq 3 \cdot 10^5$) — количество вершин и рёбер графа соответственно.

В следующих m строках находятся по три числа, обозначающих очередное ребро — u_i, v_i, w_i — номера вершин, соединённых ребром, и вес ребра ($u_i \neq v_i, 1 \leq w_i \leq 10^9$). Гарантируется, что граф связен, и что **между каждой парой вершин существует не более одного ребра**.

В последней строке ввода находится целое число u ($1 \leq u \leq n$) — номер стартовой вершины.

Выходные данные

Выведите в первой строке минимальный суммарный вес рёбер дерева.

В следующей строке выведите номера рёбер, входящих в дерево, разделяя их пробелом. Рёбра нумеруются с единицы в порядке следования во входных данных. Номера рёбер можно выводить в любом порядке.

Если возможных ответов несколько, выведите любой.

Примеры

входные данные	Скопировать
3 3 1 2 1 2 3 1 1 3 2 3	
выходные данные	Скопировать
2 1 2	

входные данные	Скопировать
4 4 1 2 1 2 3 1 3 4 1 4 1 2 4	
выходные данные	Скопировать
4 2 3 4	

Примечание

В первом тесте из условия возможны два дерева кратчайших путей:

- с ребрами 1 — 3 и 2 — 3 (суммарный вес 3);
- с ребрами 1 — 2 и 2 — 3 (суммарный вес 2);

А, например, дерево с ребрами 1 — 2 и 1 — 3 не будет деревом кратчайших путей для вершины 3, потому что расстояние от



ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

Е. Деревянная игра

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам дан лес из k корневых деревьев*. Дровосек Тимофей хочет вырубить весь лес, применяя такую операцию:

- Выбрать поддерево† какой-либо вершины одного из деревьев и удалить его из дерева.

Тимофей очень любит битовые операции, поэтому он хочет, чтобы **побитовое ИЛИ** размеров поддеревьев, которые он удалял, было максимальным. Помогите ему и найдите, какой максимальный результат он может получить.

* Деревом называется связный граф без циклов, петель и кратных ребер. Лесом называется набор из одного или нескольких деревьев. В корневом дереве одна из вершин называется корнем.

† Поддерево вершины v — это множество вершин, для которых v лежит на кратчайшем пути от этой вершины до корня, включая вершину v .

Входные данные

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

Первая строка каждого набора входных содержит одно целое число k ($1 \leq k \leq 10^6$) — количество деревьев в лесу.

Далее следует описание каждого из k деревьев:

Первая строка содержит одно целое число n ($1 \leq n \leq 10^6$) — размер дерева. Вершины дерева пронумерованы целыми числами от 1 до n . Корнем дерева является вершина под номером 1.

Вторая строка содержит $n - 1$ целое число p_2, p_3, \dots, p_n ($1 \leq p_i < i$), где p_i — предок вершины i .

Гарантируется, что сумма k и n по всем наборам входных данных не превосходит 10^6 .

Выходные данные

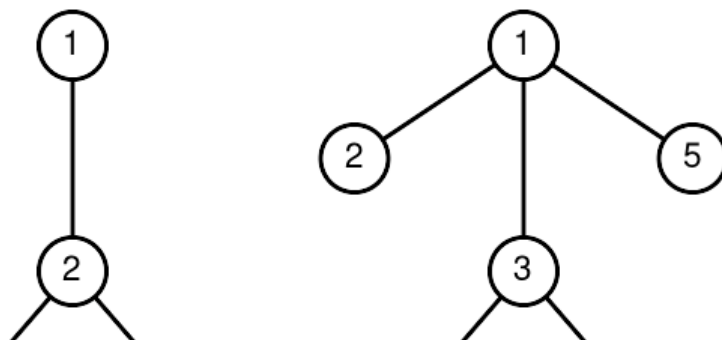
Для каждого набора входных данных выведите одно целое число — максимальный результат, который можно получить.

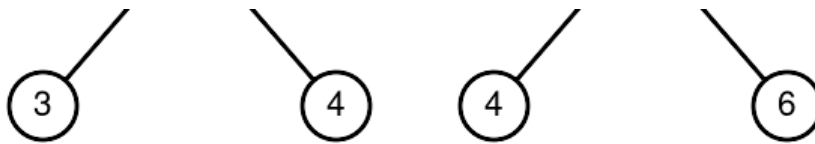
Пример

входные данные	Скопировать
3 1 1 2 4 1 2 2 6 1 1 3 1 3 1 10 1 2 2 1 1 5 7 6 4	
выходные данные	Скопировать
1 7 10	

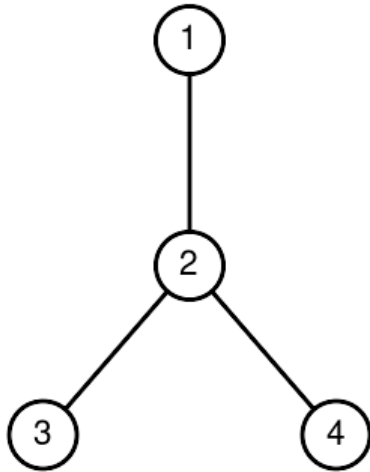
Примечание

Во втором наборе входных данных деревья выглядят так:

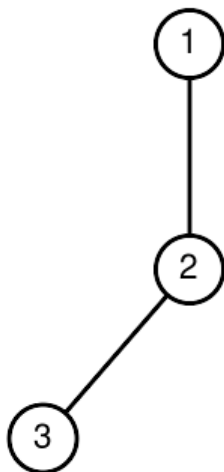




Первой операцией удалим все второе дерево.



Второй операцией удалим вершину 4 из первого дерева.



Третьей операцией удалим первое дерево. Результат равен $6|1|3 = 7$ ($|$ обозначает побитовое ИЛИ).

В третьем наборе входных данных нужно удалить все дерево.



[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Новогодний концерт

ограничение по времени на тест: 3 секунды
ограничение по памяти на тест: 256 мегабайт

Новый год совсем близко, а это значит, что в 179-й школе уже полным ходом идет подготовка к новомуднему концерту.

Всего в школе есть n классов, пронумерованных целыми числами от 1 до n , i -й класс подготовил сценку продолжительностью a_i минут.

Иднар, как главный за проведение новогоднего концерта, знает, что если на концерте будет k сценок продолжительностями b_1, b_2, \dots, b_k минут, то зрителям станет скучно, если найдутся l и r такие, что $1 \leq l \leq r \leq k$ и $\gcd(b_l, b_{l+1}, \dots, b_{r-1}, b_r) = r - l + 1$, где $\gcd(b_l, b_{l+1}, \dots, b_{r-1}, b_r)$ обозначает [наибольший общий делитель \(НОД\)](#) чисел $b_l, b_{l+1}, \dots, b_{r-1}, b_r$.

Чтобы во время концерта зрителям не стало скучно, Иднар может сколько угодно раз (возможно, ноль) попросить t -й класс ($1 \leq t \leq k$) сделать другую сценку продолжительностью d , где d может быть [любым целым положительным числом](#). Таким образом, после данной операции b_t станет равно d . Заметьте, что t и d для разных операций могут быть различными.

Для последовательности продолжительностей сценок b_1, b_2, \dots, b_k определим функцию $f(b)$ как минимальное количество классов, у которых Иднар должен попросить заменить сценку, чтобы зрителям не стало скучно.

Иднар еще точно не определился, какие сценки будут допущены на концерт, поэтому хочет узнать значение функции f от каждого из непустых префиксов a . Иными словами, Иднар хочет узнать значения $f(a_1), f(a_1, a_2), \dots, f(a_1, a_2, \dots, a_n)$.

Входные данные

Первая строка содержит единственное целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество классов в школе.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — продолжительности сценок классов.

Выходные данные

В единственной строке через пробел выведите последовательность из n чисел — $f(a_1), f(a_1, a_2), \dots, f(a_1, a_2, \dots, a_n)$.

Примеры

входные данные	Скопировать
1 1	
выходные данные	Скопировать
1	

входные данные	Скопировать
3 1 4 2	
выходные данные	Скопировать
1 1 2	

входные данные	Скопировать
7 2 12 4 8 18 3 6	
выходные данные	Скопировать
0 1 1 1 2 2 2	

Примечание

В первом тесте можно заменить 1 на 2, поэтому ответ 1.

Во втором тесте:

- [1] можно преобразовать в [2],
- [1, 4] можно преобразовать в [3, 4],
- [1, 4, 2] можно преобразовать в [2, 3, 2].

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

G. Подсчет графов

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Дано дерево, состоящее из n вершин. Дерево — это связный неориентированный граф без циклов. Каждое ребро дерева имеет свой вес, w_i .

Ваша задача — подсчитать количество различных графов, для которых одновременно выполняются четыре условия:

1. В графе нет петель и кратных ребер.
2. Веса на ребрах графа целые числа и не превышают S .
3. Граф имеет **ровно одно минимальное остовное дерево**.
4. Минимальное остовное дерево графа является заданным деревом.

Два графа считаются различными, если их наборы рёбер различны с учётом весов рёбер.

Ответ может быть большим, выведите его по модулю 998244353.

Входные данные

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

Первая строка каждого набора содержит два целых числа n и S ($2 \leq n \leq 2 \cdot 10^5, 1 \leq S \leq 10^9$) — количество вершин и верхняя граница весов.

Затем следуют $n - 1$ строк, описывающих дерево, i -я строка содержит три целых числа u_i, v_i и w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq S$) — ребро в дереве с весом w_i .

Гарантируется, что сумма n для всех тестов не превышает $2 \cdot 10^5$.

Выходные данные

Для каждого теста выведите количество различных графов, удовлетворяющих условиям, по модулю 998244353.

Пример

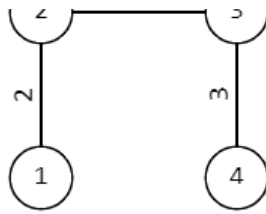
входные данные	Скопировать
4 2 5 1 2 4 4 5 1 2 2 2 3 4 3 4 3 5 6 1 2 3 1 3 2 3 4 6 3 5 1 10 200 1 2 3 2 3 33 3 4 200 1 5 132 5 6 1 5 7 29 7 8 187 7 9 20 7 10 4	
выходные данные	Скопировать
1 8 80 650867886	

Примечание

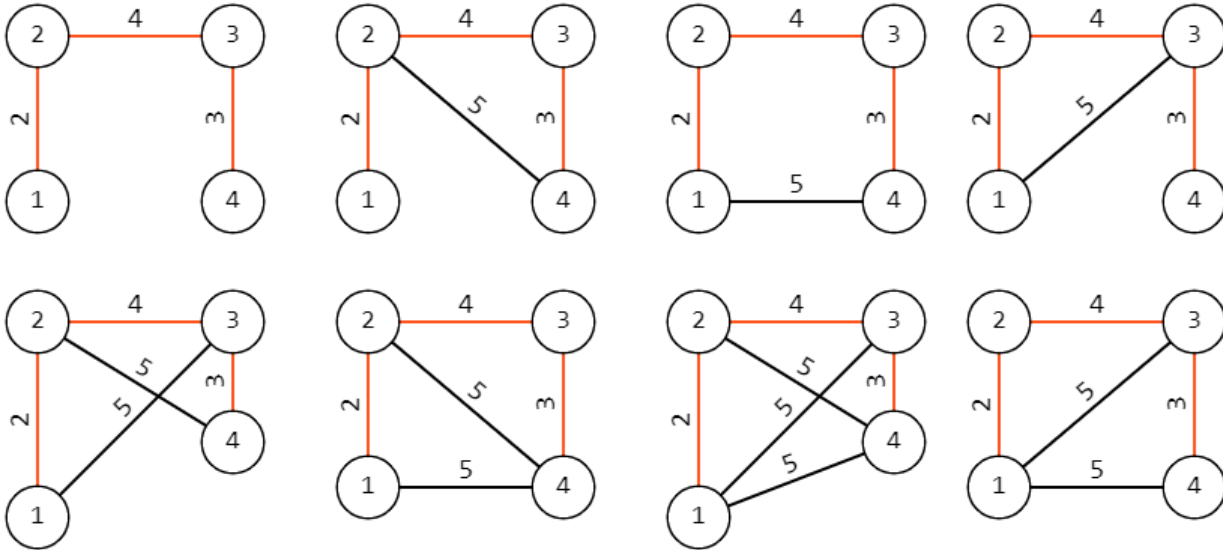
В первом примере существует единственный граф, который и является заданным деревом.

Во втором примере заданное дерево выглядит так:





Ниже изображены все возможные графы для второго примера, красным цветом выделено минимальное остовное дерево:



[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 26.07.2025 11:31:29^{UTC+5} (i1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

D. Размещение

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 512 мегабайт

Энни — фотограф-любитель. Ей нравится фотографировать гигантские жилые дома ночью. Она только что сфотографировала огромное прямоугольное здание, которое можно представить как таблицу из $n \times m$ окон. Это означает, что в здании n этажей и на каждом этаже ровно m окон. Каждое окно либо тёмное, либо светлое, то есть в комнате за ним горит свет.

Энни знает, что каждая квартира в этом доме либо однокомнатная, либо двухкомнатная. Каждая однокомнатная квартира представлена на фотографии ровно одним окном, а каждая двухкомнатная квартира представлена на фотографии двумя **подряд идущими** окнами на одном этаже. Более того, m гарантированно делится на 4 и известно, что на каждом этаже ровно $\frac{m}{4}$ двухкомнатных квартир и ровно $\frac{m}{2}$ однокомнатных квартир. Точное расположение квартир неизвестно и может быть разным для каждого этажа.

Энни считает, что квартира заселена, если **хотя бы в одном** из её окон горит свет. Теперь она задаётся вопросом, какое может быть минимальное и максимальное количество заселённых квартир в доме, если судить по данной фотографии?

Формально, для каждого из этажей Энни придумывает возможное расположение квартир, в котором есть ровно $\frac{m}{4}$ двухкомнатных квартир (два последовательных окна) и $\frac{m}{2}$ однокомнатных квартир (одно окно). Затем она подсчитывает общее количество квартир, в которых есть хотя бы одно светлое окно. Какое минимальное и максимальное число она может получить?

Входные данные

Первая строка входных данных содержит два положительных целых числа n и m ($1 \leq n \cdot m \leq 5 \cdot 10^5$) — количество этажей в здании и количество окон на этаже соответственно. Гарантируется, что m делится на 4.

Затем следует n строк, каждая из которых содержит m символов. j -й символ i -й строки равен «0», если j -е окно на i -м этаже тёмное, и равен «1», если это окно светлое.

Выходные данные

Выведите два целых числа, минимальное и максимальное возможное количество заселённых квартир, если на каждом этаже возможно своё собственное расположение $\frac{m}{4}$ двухкомнатных и $\frac{m}{2}$ однокомнатных квартир.

Примеры

входные данные	Скопировать
5 4 0100 1100 0110 1010 1011	
выходные данные	Скопировать
7 10	

входные данные	Скопировать
1 8 01011100	
выходные данные	Скопировать
3 4	

Примечание

В первом примере каждый этаж состоит из одной двухкомнатной квартиры и двух однокомнатных квартир.

При следующей планировке квартир достигается минимально возможное количество заселённых квартир, равное 7.

```
0 1 0 0 |
1 1 0 0 |
0 1 1 0 |
1 0 1 0 |
1 0 1 1 |
```

При следующей планировке квартир достигается максимально возможное количество заселённых квартир, равное 10.

```
|0 1|0|0|
|1|1 0|0|
|0 1|1|0|
|1|0 1|0|
|1 0|1|1|
```

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 26.07.2025 11:31:31^{UTC+5} (i1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

D. Омкар и медианы

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

О-о-о! Рэй снова потерял свой массив! Однако Омкар может помочь, потому что он думает, что нашел *OmkaArray* массива Рэя. *OmkaArray* массива a с элементами $a_1, a_2, \dots, a_{2k-1}$ — это массив b с элементами b_1, b_2, \dots, b_k такой, что b_i равен медиане $a_1, a_2, \dots, a_{2i-1}$ для всех i . Омкар нашел массив b размера n ($1 \leq n \leq 2 \cdot 10^5$, $-10^9 \leq b_i \leq 10^9$). Для этого массива b , Рэй хочет проверить утверждение Омкара и узнать, действительно ли b является *OmkaArray* некоторого массива a . Можете ли вы помочь Рэю?

Медиана набора чисел $a_1, a_2, \dots, a_{2i-1}$ — это число c_i , где $c_1, c_2, \dots, c_{2i-1}$ представляют собой $a_1, a_2, \dots, a_{2i-1}$, отсортированные в неубывающем порядке.

Входные данные

Каждый тест содержит несколько 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 ($1 \leq n \leq 2 \cdot 10^5$) — длину массива b .

Вторая строка содержит n целых чисел b_1, b_2, \dots, b_n ($-10^9 \leq b_i \leq 10^9$) — элементы b .

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

Выходные данные

Для каждого набора входных данных выведите одну строку, содержащую YES, если существует массив a такой, что b_i является медианой $a_1, a_2, \dots, a_{2i-1}$ для всех i , и NO в противном случае. Регистр букв в YES и NO не имеет значения (поэтому yES и No также будут приняты).

Примеры

входные данные	Скопировать
5 4 6 2 1 3 1 4 5 4 -8 5 6 -7 2 3 3 4 2 1 2 3	
выходные данные	Скопировать
NO YES NO YES YES	

входные данные	Скопировать
5 8 -8 2 -6 -5 -4 3 3 2 7 1 1 3 1 0 -2 -1 7 6 12 8 6 2 6 10 6 5 1 2 3 6 7 5 1 3 4 3 0	
выходные данные	Скопировать
NO YES NO NO NO	

Примечание

Во втором наборе входных данных первого примера массив $[4]$ даст `OmkArray` $[4]$, так как медиана первого элемента равна 4.

В четвертом наборе входных данных первого примера массив $[3, 2, 5]$ даст `OmkArray` $[3, 3]$, так как медиана 3 равна 3, а медиана 2, 3, 5 равна 3.

В пятом наборе входных данных первого примера массив $[2, 1, 0, 3, 4, 4, 3]$ даст `OmkArray` $[2, 1, 2, 3]$, так как

- медиана 2 равна 2
- медиана 0, 1, 2 равна 1
- медиана 0, 1, 2, 3, 4 равна 2
- и медиана 0, 1, 2, 3, 3, 4, 4 равна 3.

Во втором наборе входных данных второй выборки массив $[1, 0, 4, 3, 5, -2, -2, -2, -4, -3, -4, -1, 5]$ даст `OmkArray` $[1, 1, 3, 1, 0, -2, -1]$, как

- медиана 1 равна 1
- медиана из 0, 1, 4 равна 1
- медиана из 0, 1, 3, 4, 5 равна 3
- медиана $-2, -1, 0, 1, 3, 4, 5$ равна 1
- медиана $-4, -2, -2, -2, 0, 1, 3, 4, 5$ равна 0
- медиана $-4, -4, -3, -2, -2, -2, 0, 1, 3, 4, 5$ равна -2
- и медиана $-4, -4, -3, -2, -2, -2, -1, 0, 1, 3, 4, 5, 5$ равна -1

Для всех случаев, когда ответ NO, можно доказать, что невозможно найти массив a такой, что b является `Omkarray` a .

[Codeforces](#) (c) Copyright 2010-2025 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 26.07.2025 11:31:33^{UTC+5} (i1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#) | [Terms and Conditions](#)

При поддержке



ИТМО

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Максимальный подмассив

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

Задан массив a_1, a_2, \dots, a_n , состоящий из n целых чисел. Также дано два целых числа k и x .

Вы должны выполнить следующую операцию ровно один раз: прибавить x к элементам на **ровно k различных** позициях и вычесть x на всех остальных.

Например, если $a = [2, -1, 2, 3]$, $k = 1$, $x = 2$, и мы выбрали увеличить первый элемент, тогда после применения операции $a = [4, -3, 0, 1]$.

Пусть $f(a)$ — максимальная сумма подмассива из a . Подмассив массива a — это последовательная часть массива a , другими словами массив a_i, a_{i+1}, \dots, a_j для некоторых $1 \leq i \leq j \leq n$. Пустой подмассив тоже рассматривается, его сумма равна 0.

Пусть массив a' будет массивом a после применения вышеупомянутой операции. Примените операцию таким образом, чтобы $f(a')$ было максимально возможным, и выведите максимально возможное значение $f(a')$.

Входные данные

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

Первая строка каждого набора содержит три целых числа n , k и x ($1 \leq n \leq 2 \cdot 10^5$; $0 \leq k \leq \min(20, n)$; $-10^9 \leq x \leq 10^9$).

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

Сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого набора входных данных выведите одно число — максимально возможное значение $f(a')$.

Пример

входные данные	Скопировать
4 4 1 2 2 -1 2 3 2 2 3 -1 2 3 0 5 3 2 4 6 2 -8 4 -1 9 -3 7 -8	
выходные данные	Скопировать
5 7 0 44	



ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

Е. Дешевый обед

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 512 мегабайт

Иван хочет полноценно пообедать. Для этого он хочет заказать первое, второе, напиток и десерт.

Всего на выбор у Ивана есть n_1 разных первых блюд (i -е первое стоит a_i монет), n_2 разных вторых блюд (i -е второе стоит b_i монет), n_3 разных напитков (i -й напиток стоит c_i монет) и n_4 разных десертов (i -й десерт стоит d_i монет).

Некоторые блюда не сочетаются между собой. Всего есть m_1 пар из первого и второго, которые не сочетаются, m_2 несочетающихся пар из второго и напитка и m_3 несочетающихся пар из напитка и десерта.

Иван хочет купить ровно одно первое, одно второе, один напиток и один десерт так, чтобы они хорошо сочетались между собой и их общая стоимость была наименьшей возможной. Помогите ему найти самый бюджетный вариант!

Входные данные

В первой строке заданы четыре целых числа n_1, n_2, n_3 и n_4 ($1 \leq n_i \leq 150000$) — количество видов первого, второго, напитков и десертов соответственно.

Далее следуют четыре строки. В первой строке заданы n_1 целых чисел a_1, a_2, \dots, a_{n_1} ($1 \leq a_i \leq 10^8$), где a_i — это цена i -го вида первого блюда. Следующие три строки описывают цены вторых блюд, напитков и десертов в том же формате ($1 \leq b_i, c_i, d_i \leq 10^8$).

В следующей строке задан одно целое число m_1 ($0 \leq m_1 \leq 200000$) — количество несочетающихся пар из первого и второго блюд. В каждой из следующих m_1 строк заданы два целых числа x_i и y_i ($1 \leq x_i \leq n_1; 1 \leq y_i \leq n_2$) означающие, что первое блюдо номер x_i не сочетается со вторым блюдом номер y_i . Все пары различные.

Блок несочетающихся пар из второго блюда и напитка задан в аналогичном формате. В таком же формате заданы и несочетающиеся пары из напитка и десерта ($0 \leq m_2, m_3 \leq 200000$).

Выходные данные

Если невозможно выбрать первое, второе, напиток и десерт, сочетающиеся друг с другом, выведите -1 . В противном случае выведите одно число — наименьшую суммарную стоимость обеда.

Примеры

входные данные	Скопировать
4 3 2 1 1 2 3 4 5 6 7 8 9 10 2 1 2 1 1 2 3 1 3 2 1 1 1	
выходные данные	Скопировать
26	

входные данные	Скопировать
1 1 1 1 1 1 1 1 1 1 1 0 0	
выходные данные	Скопировать
-1	

Примечание

ЗАДАЧИ ОТОСЛАТЬ СТАТУС ПОЛОЖЕНИЕ ЗАПУСК

D. Берсерк и Огненный Шар

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

В ряд стоят n воинов. Сила i -го воина равна a_i . Все силы различны.

Вы можете использовать заклинания двух типов:

- Огненный Шар: вы тратите x маны и уничтожаете **ровно** k последовательных воинов;
- Берсерк: вы тратите y маны, выбираете двух соседних воинов, и воин с большей силой убивает воина с меньшей силой.

Например, пусть силы воинов равны $[2, 3, 7, 8, 11, 5, 4]$, и $k = 3$. Если использовать Берсерк на воинах с силой 8 и 11, последовательность станет $[2, 3, 7, 11, 5, 4]$. Если после этого использовать Огненный Шар на воинах с силами $[7, 11, 5]$, последовательность станет $[2, 3, 4]$.

Вам необходимо превратить последовательность сил воинов a_1, a_2, \dots, a_n в b_1, b_2, \dots, b_m . Посчитайте минимальное количество маны, которое вы должны затратить.

Входные данные

Первая строка содержит два числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — длина последовательности a и длина последовательности b соответственно.

Вторая строка содержит три числа x, k, y ($1 \leq x, y \leq 10^9; 1 \leq k \leq n$) — стоимость Огненного Шара, длина действия Огненного Шара и стоимость Берсерка соответственно.

Третья строка содержит n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$). Гарантируется что числа a_i попарно различны.

Четвертая строка содержит m чисел b_1, b_2, \dots, b_m ($1 \leq b_i \leq n$). Гарантируется что числа b_i попарно различны.

Выходные данные

Выведите минимальное количество маны для превращения последовательности a_1, a_2, \dots, a_n в b_1, b_2, \dots, b_m (или -1 , если это невозможно).

Примеры

входные данные	Скопировать
5 2 5 2 3 3 1 4 5 2 3 5	
выходные данные	Скопировать
8	
входные данные	Скопировать
4 4 5 1 4 4 3 1 2 2 4 3 1	
выходные данные	Скопировать
-1	
входные данные	Скопировать
4 4 2 1 11 1 3 2 4 1 3 2 4	
выходные данные	Скопировать
0	

[ЗАДАЧИ](#) [ОТОСЛАТЬ](#) [СТАТУС](#) [ПОЛОЖЕНИЕ](#) [ЗАПУСК](#)

D. Пары отрезков

ограничение по времени на тест: 4 секунды
ограничение по памяти на тест: 512 мегабайт

Два отрезка $[l_1, r_1]$ и $[l_2, r_2]$ пересекаются, если существует хотя бы одно число x , такое, что $l_1 \leq x \leq r_1$ и $l_2 \leq x \leq r_2$.

Массив отрезков $[[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]]$ называется **красивым**, если k чётно, и его можно разбить на $\frac{k}{2}$ пар таким образом, что:

- каждый элемент массива принадлежит ровно одной паре;
- отрезки в каждой паре пересекаются между собой;
- отрезки из разных пар не пересекаются.

Например, массив $[[2, 4], [9, 12], [2, 4], [7, 7], [10, 13], [6, 8]]$ является красивым, так как его можно разбить на 3 пары следующим образом:

- первый элемент массива (отрезок $[2, 4]$) и третий элемент массива (отрезок $[2, 4]$);
- второй элемент массива (отрезок $[9, 12]$) и пятый элемент массива (отрезок $[10, 13]$);
- четвёртый элемент массива (отрезок $[7, 7]$) и шестой элемент массива (отрезок $[6, 8]$).

Как видно, отрезки в каждой паре пересекаются, а отрезки из разных пар не пересекаются.

Дан массив из n отрезков $[[l_1, r_1], [l_2, r_2], \dots, [l_n, r_n]]$. Вам нужно удалить минимально возможное количество элементов из этого массива, чтобы получившийся массив был красивым.

Входные данные

Первая строка содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($2 \leq n \leq 2000$) — количество отрезков в массиве. Затем следуют n строк, i -я из которых содержит два целых числа l_i и r_i ($0 \leq l_i \leq r_i \leq 10^9$), обозначающих i -й отрезок.

Дополнительное ограничение на входные данные: сумма n по всем наборам входных данных не превышает 2000.

Выходные данные

Для каждого набора входных данных выведите одно целое число — минимальное количество элементов, которые нужно удалить, чтобы получившийся массив был красивым.

Пример

входные данные	Скопировать
3 7 2 4 9 12 2 4 7 7 4 8 10 13 6 8 5 2 2 2 8 0 10 1 2 5 6 4 1 1 2 2 3 3 4 4	
выходные данные	Скопировать
1 3 4	

Примечание

C. Очередная задача про массив

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт

Вам дан массив a из n целых чисел. Вы можете совершить следующую операцию любое число раз (0 или больше раз):

- Выбрать 2 индекса i, j где $1 \leq i < j \leq n$ и заменить a_k для всех $i \leq k \leq j$ значением $|a_i - a_j|$

Выведите максимальную сумму всех элементов конечного массива, которую вы можете получить таким образом.

Входные данные

В первой строке содержится одно целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных.

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

Вторая строка каждого набора содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива a .

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

Выходные данные

Для каждого набора входных данных выведите сумму конечного массива.

Пример

входные данные	Скопировать
3 3 1 1 1 2 9 1 3 4 9 5	
выходные данные	Скопировать
3 16 18	

Примечание

В первом примере невозможно достичь сумму > 3 используя операцию, таким образом ответ 3.

Во втором примере можно показать, что максимальная достижимая сумма равна 16. Используем операцию (1, 2) и преврати массив из $[9, 1]$ в $[8, 8]$, таким образом ответ равен 16.

В третьем примере можно показать, что невозможно достичь суммы > 18 используя операцию, таким образом ответ 18.

