

А. Интересная сумма

1 секунда, 256 мегабайт

Дан массив a длины n . Вы можете выбрать любой подотрезок a_l, a_{l+1}, \dots, a_r массива длины, не совпадающий со всем массивом, то есть, для которого $1 \leq l \leq r \leq n$ и $r - l + 1 < n$. Красотой выбранного подотрезка назовем значение

$$\max(a_1, a_2, \dots, a_{l-1}, a_{r+1}, a_{r+2}, \dots, a_n) - \min(a_1, a_2, \dots, a_{l-1}, a_{r+1}, a_{r+2}, \dots, a_n)$$

Найдите максимальную красоту подотрезка среди всех возможных допустимых подотрезков массива (за исключением всего массива).

Входные данные

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

Первая строка каждого набора входных данных содержит единственное целое число n ($4 \leq n \leq 10^5$) — длину массива.

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

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

Выходные данные

Для каждого набора входных данных выведите одно число — искомое максимальное значение красоты.

входные данные
4
8
1 2 2 3 1 5 6 1
5
1 2 3 100 200
4
3 3 3 3
6
7 8 3 1 1 8
выходные данные
9
297
0
14

В первом тесте из условия оптимально выбрать отрезок $l = 7$, $r = 8$. Красота этого отрезка равна $(6 - 1) + (5 - 1) = 9$.

Во втором тесте из условия оптимально выбрать отрезок $l = 2$, $r = 4$. Красота этого отрезка равна $(100 - 2) + (200 - 1) = 297$.

В. Фотограф Марк

1 секунда, 256 мегабайт

Марка попросили сфотографировать группу из $2n$ человек, i -й из них имеет рост h_i .

Для этого он выстроил этих людей в два ряда: передний и задний, каждый из которых состоит из n человек. На фотографии все люди хорошо видны, если рост j -го человека в дальнем ряду как минимум на x больше, чем рост j -го человека в ближнем ряду для каждого j от 1 до n включительно.

Помогите Марку определить, возможно ли расставить людей описанным выше способом.

Входные данные

Первая строка входных данных содержит целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Каждый набор входных данных состоит из двух строк.

Первая строка каждого набора содержит два положительных целых числа n и x ($1 \leq n \leq 100$, $1 \leq x \leq 10^3$) — количество людей в каждом ряду и минимальную разницу в росте, которую хочет получить Марк.

Вторая строка каждого набора содержит $2n$ положительных целых чисел h_1, h_2, \dots, h_{2n} ($1 \leq h_i \leq 10^3$) — рост каждого из людей.

Обратите внимание, что сумма значений n по всем наборам входных данных не превышает 1000.

Выходные данные

Для каждого набора входных данных выведите «YES», если Марк может расставить людей в два ряда требуемым образом. В противном случае выведите «NO».

Вы можете выводить «YES» и «NO» в любом регистре (например, строки «yES», «yes» и «Yes» будут распознаны как правильный ответ).

входные данные
3
3 6
1 3 9 10 12 16
3 1
2 5 2 2 2 5
1 2
8 6
выходные данные
YES
NO
YES

В первом наборе входных данных из примера один из возможных требуемых порядков имеет следующий вид: третий, пятый и шестой человек должны встать в дальний ряд, а второй, первый и четвертый — в ближний ряд. В этом случае расположение будет выглядеть так:

Дальний ряд	9	12	16
Ближний ряд	3	1	10

Оно соответствует требованиям, так как:

- $h_3 - h_2 = 9 - 3 \geq 6$,
- $h_5 - h_1 = 12 - 1 \geq 6$, and
- $h_6 - h_4 = 16 - 10 \geq 6$.

Для второго набора входных данных может быть показано, что требуемого расположения людей не существует.

В третьем наборе входных данных единственный подходящий способ расположения следующий: второй человек стоит в дальнем ряду, а первый в ближнем.

С. Все различные

1 секунда, 256 мегабайт

У Sho есть массив a состоящий из n целых чисел. За одну операцию Sho может выбрать два **различных** индекса i и j , после чего удалить из массива элементы a_i и a_j .

Например, для массива $[2, 3, 4, 2, 5]$ Sho может выбрать индексы 1 и 3 и удалить соответствующие элементы из массива. После этой операции массив будет выглядеть так: $[3, 2, 5]$. Заметьте, что после любой операции длина массива уменьшится на два.

После нескольких операций у Sho остался массив, содержащий только **различные** числа. Также, он применил операции таким образом, что длина оставшегося массива **максимальна** из всех возможных.

Более формально, после всех операций массив Sho удовлетворяет двум следующим критериям:

- В массиве не существует таких пар индексов, что $i < j$ и $a_i = a_j$.
- Длина массива a максимальна.

Выведите длину оставшегося у Sho массива.

Входные данные

Первая строка входных данных содержит целое число t ($1 \leq t \leq 10^3$) — количество наборов входных данных.

Первая строка каждого набора данных содержит единственное число n ($1 \leq n \leq 50$) — длину начального массива.

Вторая строка каждого набора данных содержит n целых чисел a_i ($1 \leq a_i \leq 10^4$) — элементы начального массива.

Выходные данные

Для каждого набора данных выведите единственное число — длину оставшегося массива. Помните, что в оставшемся массиве все элементы различны, а его длина максимальна.

входные данные
4
6
2 2 2 3 3 3
5
9 1 9 9 1
4
15 16 16 15
4
10 100 1000 10000
выходные данные
2
1
2
4

В первом наборе данных Sho может сделать следующие операции:

- Выбрать индексы 1 и 5. Тогда массив станет $[2, 2, 2, 3, 3, 3] \rightarrow [2, 2, 3, 3]$.
- Выбрать индексы 1 и 4. Тогда массив станет $[2, 2, 3, 3] \rightarrow [2, 3]$.

Финальный массив имеет длину 2, так что ответом является 2. Можно доказать, что Sho не может получить массив большей длины.

Во втором наборе данных Sho может сделать следующие операции:

- Выбрать индексы 3 и 4. Тогда массив станет $[9, 1, 9, 9, 1] \rightarrow [9, 1, 1]$.
- Выбрать индексы 1 и 3. Тогда массив станет $[9, 1, 1] \rightarrow [1]$.

Финальный массив имеет длину 1, так что ответом является 1. Можно доказать, что Sho не может получить массив большей длины.

D. Тройка

1 секунда, 256 мегабайт

Вам задан массив a из n элементов. Найдите любое число, которое встречается в этом массиве хотя бы три раза, или выведите -1 , если таких чисел нет.

Входные данные

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

Первая строка набора входных данных содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество элементов в массиве.

Вторая строка набора входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) — элементы массива.

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

Выходные данные

Для каждого набора входных данных выведите любое целое число, которое встречается в массиве хотя бы три раза, или -1 , если такого числа нет.

входные данные
7
1
1
3
2 2 2
7
2 2 3 3 4 2 2
8
1 4 3 4 3 2 4 1
9
1 1 1 2 2 2 3 3 3
5
1 5 2 4 3
4
4 4 4 4
выходные данные
-1
2
2
4
3
-1
4

В первом наборе входных данных в массиве всего один элемент, поэтому не существует числа, которое встречается хотя бы три раза. Следовательно, ответ равен -1 .

Во втором наборе входных данных все три элемента массива равны 2, поэтому 2 встречается трижды. Следовательно, в качестве ответа надо вывести 2.

В третьем наборе входных данных 2 встречается четыре раза, поэтому 2 — ответ.

В четвертом наборе входных данных 4 встречается три раза, поэтому 4 — ответ.

В пятом наборе входных данных каждое из чисел 1, 2 и 3 встречается хотя бы три раза, поэтому любое из них может быть ответом.

В шестом наборе входных данных все элементы массива различны, поэтому ни одно число не встречается хотя бы три раза, и ответ равен -1 .

E. Сортировка нечётными обменами

1 секунда, 256 мегабайт

Вам дан массив a_1, a_2, \dots, a_n . Вы можете выполнять операции с массивом. За одну операцию вы можете выбрать целое число i ($1 \leq i < n$) и поменять местами a_i и a_{i+1} в массиве, если $a_i + a_{i+1}$ нечётно.

Определите, можно ли его отсортировать в неубывающем порядке, используя эти операции любое количество раз.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится единственное целое число t ($1 \leq t \leq 10^5$) — количество наборов входных данных. Далее следует описание наборов входных данных.

В первой строке каждого набора входных данных содержится целое число n ($1 \leq n \leq 10^5$) — длина массива.

Во второй строке каждого набора входных данных содержатся n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива.

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

Выходные данные

Для каждого набора входных данных выведите «Yes» или «No» в зависимости от того, можете ли вы или нет отсортировать данный массив.

Вы можете выводить каждую букву в любом регистре (например, «YES», «Yes», «yes», «yEs») будут распознаны как положительный ответ).

входные данные
4
4
1 6 31 14
2
4 2
5
2 9 6 7 10
3
6 6 6
выходные данные
Yes
No
No
Yes

В первом наборе входных данных мы можем просто поменять местами 31 и 14 ($31 + 14 = 45$, что нечётно) и получить неубывающий массив $[1, 6, 14, 31]$.

Во втором наборе входных данных единственным способом отсортировать массив является поменять местами 4 и 2, но это невозможно, поскольку их сумма $4 + 2 = 6$ является чётной.

В третьем наборе входных данных мы не можем сделать последовательность неубывающей.

В четвёртом наборе входных данных массив уже является неубывающим.

Ф. Полуфиналы

1 секунда, 256 мегабайт

В турнире по бегу только что прошли два полуфинала. В каждом полуфинале участвовало n человек. В финал проходят n человек, определяемых следующим образом: из каждого полуфинала выбирают k человек ($0 \leq 2k \leq n$), показавших наилучший результат в своем полуфинале, а все остальные места в финале достаются тем, кто не попал в первые k в своем полуфинале, но попал в число $n - 2k$ лучших среди остальных.

Организаторы турнира пока не определили число k , поэтому участники хотят знать, у кого еще остались шансы попасть в финал, а кому уже можно отправляться домой.

Входные данные

В первой строке записано единственное целое число n ($1 \leq n \leq 10^5$) — количество участников в каждом полуфинале.

В следующих n строках записано по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq 10^9$) — результаты i -ого участника (количество миллисекунд, за которое он пробежал дистанцию полуфинала) первого и второго полуфиналов соответственно. Все результаты различны. Последовательности a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n упорядочены по возрастанию — в том порядке, в каком участники финишировали в соответствующем полуфинале.

Выходные данные

Выведите две строки, состоящие n символов, каждый из которых — «0» или «1». Первая строка должна соответствовать участникам первого полуфинала, а вторая — участникам второго полуфинала. i -ый символ в j -ой строке должен быть равен «1», если i -ый участник j -ого полуфинала имеет шансы пройти в финал, и «0» — иначе.

входные данные
4
9840 9920
9860 9980
9930 10020
10040 10090
выходные данные
1110
1100

входные данные
4
9900 9850
9940 9930
10000 10020
10060 10110
выходные данные
1100
1100

Рассмотрим первый пример. В каждом полуфинале участвовало 4 человека. Результаты первого полуфинала — 9840, 9860, 9930, 10040. Результаты второго полуфинала — 9920, 9980, 10020, 10090.

- В случае $k = 0$ финалисты определяются исключительно по времени, поэтому дальше пройдут спортсмены с результатами 9840, 9860, 9920 и 9930.
- В случае $k = 1$ из обоих полуфиналов гарантированно проходят победители (с результатами 9840 и 9920), а оставшиеся места определяются по времени (эти места достанутся спортсменам, пробежавшим за 9860 и 9930 миллисекунд).
- В случае $k = 2$ из обоих полуфиналов проходят по два первых места, это спортсмены с результатами 9840, 9860, 9920 и 9980 миллисекунд.

Г. Таблица Юнга

2 секунды, 256 мегабайт

Вам задана таблица a , состоящая из n строк, пронумерованных от 1 до n . В i -ой строке таблицы a содержится c_i клеточек, при этом для всех i ($1 < i \leq n$) выполняется $c_i \leq c_{i-1}$.

Обозначим за s общее количество клеточек таблицы a , то есть $s = \sum_{i=1}^n c_i$. Известно, что в каждой клеточке таблицы a записано единственное целое число от 1 до s , при этом все записанные числа различны.

Пусть клеточки i -той строки таблицы a пронумерованы от 1 до c_i , тогда обозначим число, записанное в j -той клеточке i -той строки, за $a_{i,j}$. Вам необходимо посредством нескольких операций обмена переупорядочить числа в таблице так, чтобы выполнялись следующие условия:

- для всех i, j ($1 < i \leq n; 1 \leq j \leq c_i$) выполняется $a_{i,j} > a_{i-1,j}$;
- для всех i, j ($1 \leq i \leq n; 1 < j \leq c_i$) выполняется $a_{i,j} > a_{i,j-1}$.

За одну операцию обмена разрешается выбрать две различных клеточки таблицы и поменять записанные в них числа местами, то есть число, которое до применения операции было записано в первой из выбранных клеток, после применения операции будет записано во второй. Аналогично, число, которое до применения операции было записано во второй из выбранных клеток, после применения операции будет записано в первой.

Переупорядочите числа требуемым способом. Учтите, что Вам разрешается выполнить любое количество операций не превосходящее s . Минимизировать количество операций не требуется.

Входные данные

В первой строке записано единственное целое число n ($1 \leq n \leq 50$), обозначающее количество строк в таблице. Во второй строке через пробел записаны n целых чисел c_i ($1 \leq c_i \leq 50$; $c_i \leq c_{i-1}$) — количества клеточек в соответствующих строках.

В следующих n строках задана таблица a . В i -той из них через пробел записаны c_i целых чисел: j -тое число в этой строке обозначает $a_{i,j}$.

Гарантируется, что все заданные числа $a_{i,j}$ положительны и не превосходят s . Гарантируется, что все $a_{i,j}$ различны.

Выходные данные

В первой строке выведите единственное целое число m ($0 \leq m \leq s$), обозначающее количество произведенных операций обмена.

В следующих m строках выведите описание этих операций обмена. В i -той из них выведите через пробел четыре целых числа x_i, y_i, p_i, q_i ($1 \leq x_i, p_i \leq n$; $1 \leq y_i \leq c_{x_i}$; $1 \leq q_i \leq c_{p_i}$). Выведенные числа обозначают операцию обмена содержимого клеток a_{x_i, y_i} и a_{p_i, q_i} . Обратите внимание, что операцией обмена можно менять содержимое **различных** клеток таблицы. Выводите обмены в том порядке, в котором они должны производиться.

входные данные
3 3 2 1 4 3 5 6 1 2
выходные данные
2 1 1 2 2 2 1 3 1

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

Н. Отчёт

2 секунды, 256 мегабайт

Каждый месяц Блейк получает отчёт об основных показателях деятельности компании «Blake Technologies». Компания производит n продуктов, поэтому в отчёте указано n целых чисел — выручка по каждому из этих продуктов. Однако перед тем, как этот отчёт попадёт в руки Блейка, он проходит нелёгкий путь через m менеджеров. Каждый из менеджеров, прежде чем передать отчёт следующему менеджеру (или Блейку), может переставить некоторые числа по своему усмотрению. А именно: i -й менеджер сортирует первые r_i чисел в порядке неубывания или невозрастания, после чего передаёт отчёт менеджеру с номером $i + 1$, если $i < m$, или Блейку, если $i = m$.

Сейчас как раз настал момент, когда сотрудники составляют очередной отчёт для Блейка. Вам известна изначальная последовательность длины n , а также дано описание каждого менеджера, то есть значение r_i и какой порядок сортировки он предпочитает. Вас попросили ускорить процесс и сразу определить итоговый вид отчёта.

Входные данные

В первой строке входных данных записаны два целых числа n и m ($1 \leq n, m \leq 200\,000$) — количество чисел в отчёте и количество менеджеров соответственно.

Во второй строке содержатся n целых чисел a_i ($|a_i| \leq 10^9$) — числа в отчете, до того как он попал в руки первого менеджера.

В следующих m строках содержатся описания последовательных операций с отчётом, которые проводят менеджеры. В i -й из этих строк записаны два целых числа t_i и r_i ($t_i \in \{1, 2\}$, $1 \leq r_i \leq n$), означающих, что i -й менеджер сортирует первые r_i чисел в порядке неубывания, если $t_i = 1$, или в порядке невозрастания, если $t_i = 2$.

Выходные данные

В единственной строке выведите n целых чисел — итоговый отчёт, который попадёт к Блейку.

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

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

В первом примере изначально отчёт выглядел так: 1 2 3. После первого менеджера были переставлены первые два числа: 2 1 3. В таком виде отчёт попал к Блейку.

Во втором примере первоначально отчёт был таким: 1 2 4 3. После первого менеджера отчёт стал таким: 4 2 1 3. После второго менеджера отчёт стал выглядеть так: 2 4 1 3. Такой отчёт был передан Блейку.