

А. Удалить наименьшее

1 секунда, 256 мегабайт

Вам дан массив a , состоящий из n положительных (больших нуля) целых чисел.

За один ход вы можете выбрать два индекса i и j ($i \neq j$) таких, что абсолютная разность между a_i и a_j не превосходит единицу ($|a_i - a_j| \leq 1$), и удалить наименьший из этих двух элементов. Если два элемента равны, вы можете удалить любой из них (но только один).

Ваша задача — определить, возможно ли получить массив, состоящий **только из одного элемента**, применив несколько (возможно, ноль) таких ходов, или нет.

Вам нужно ответить на t независимых наборов тестовых данных.

Входные данные

Первая строка теста содержит одно целое число t ($1 \leq t \leq 1000$) — количество наборов тестовых данных. Затем следуют t наборов тестовых данных.

Первая строка набора тестовых данных содержит одно целое число n ($1 \leq n \leq 50$) — длину a . Вторая строка набора тестовых данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), где a_i — i -й элемент a .

Выходные данные

Для каждого набора тестовых данных выведите ответ на него: «YES», если возможно получить массив, состоящий **только из одного элемента**, применив несколько (возможно, ноль) ходов, описанных в условии задачи, или «NO» в обратном случае.

входные данные
5 3 1 2 2 4 5 5 5 5 3 1 2 4 4 1 3 4 4 1 100
выходные данные
YES YES NO NO YES

В первом наборе тестовых данных примера мы можем применить следующую последовательность ходов:

- выбрать $i = 1$ и $j = 3$, а затем удалить a_i (таким образом, a превратится в $[2; 2]$);
- выбрать $i = 1$ и $j = 2$, а затем удалить a_j (таким образом, a превратится в $[2]$).

Во втором наборе тестовых данных примера мы можем выбирать любые возможные i и j в течение любого хода и удалять любой элемент из пары.

В третьем наборе тестовых данных примера не существует способа избавиться от 2 и 4.

В. Карточки

2 секунды, 256 мегабайт

Когда Серёже было три года, ему подарили на день рождения набор карточек с буквами. С их помощью было записано словамилюбимое число мамы мальчика в двоичной системе счисления. Серёжа тотчас же принялся с ними играть, но так как не умел читать, перемешал их в случайном порядке. Папа решил привести в порядок карточки. Помогите ему восстановить исходное число при условии, что оно было максимально возможным.

Входные данные

Первая строка содержит одно целое число n ($1 \leq n \leq 10^5$) — длина строки. Вторая строка содержит строку из строчных латинских букв: «z», «e», «r», «o» и «n».

Гарантируется, что буквы можно переупорядочить таким образом, чтобы они образовывали последовательность слов, каждое из которых является либо словом «zero», что соответствует цифре 0, либо словом «one», что соответствует цифре 1.

Выходные данные

Выведите максимально возможное число в двоичной системе счисления. Выводите двоичные цифры, разделяя их пробелами. Лидирующие нули допустимы.

входные данные
4 ezor
выходные данные
0

входные данные
10 nznnooeoer
выходные данные
1 1 0

В первом примере правильная последовательность букв — это «zero».

Во втором примере правильная последовательность букв — это «oneonezero».

С. Формирование команд

1 секунда, 256 мегабайт

В университете учатся n студентов. Количество студентов чётно. Умение i -го студента программировать равно a_i .

Тренер хочет сформировать $\frac{n}{2}$ команд. Каждая команда должна состоять ровно из двух студентов, а каждый студент должен принадлежать ровно одной команде. Два студента могут сформировать команду только тогда, когда их умение программировать одинаково (иначе они не смогут понять друг друга и не смогут сформировать команду).

Студенты могут решать задачи, чтобы улучшать их умение программировать. Одна решенная задача увеличивает умение программировать на один.

Тренер хочет знать, какое минимальное количество задач студентам необходимо решить, чтобы сформировать ровно $\frac{n}{2}$ команд (то есть каждая пара студентов должна формировать команду). Ваша задача — найти это количество.

Входные данные

Первая строка входных данных содержит одно целое число n ($2 \leq n \leq 100$) — количество студентов. Гарантируется, что n всегда является чётным числом.

Вторая строка входных данных содержит n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$), где a_i равно умению i -го студента программировать.

Выходные данные

Выведите одно целое число — минимальное количество задач, которое необходимо решить студентам, чтобы сформировать ровно $\frac{n}{2}$ команд.

входные данные
6 5 10 2 3 14 5
выходные данные
5

входные данные
2 1 100
выходные данные
99

В первом тестовом примере оптимальными будут команды: (3, 4), (1, 6) и (2, 5), где числа в скобках являются номерами студентов. Тогда для того, чтобы сформировать первую команду, третий студент должен решить 1 задачу, чтобы сформировать вторую команду, никто не должен решать задачи, и чтобы сформировать третью команду, второй студент должен решить 4 задачи, таким образом ответ равен $1 + 4 = 5$.

Во втором тестовом примере первый студент должен решить 99 задач, чтобы сформировать команду со вторым.

D. Денежная система Геральдиона

2 секунды, 256 мегабайт

На волшебном острове Геральдионе, на котором живёт Геральд, есть своя денежная система. В обращении находятся купюры нескольких номиналов. Но, к сожалению, их система несовершенна, и иногда Геральдионцы сталкиваются с ситуацией, что определённую сумму денег нельзя выдать без сдачи никаким набором купюр (естественно, можно использовать сколько угодно купюр каждого достоинства). Такая сумма называется *неудачной*. Геральд заинтересовался: а какова минимальная *неудачная* сумма?

Входные данные

В первой строке дано число n ($1 \leq n \leq 1000$) — количество номиналов купюр, которые имеют хождение на Геральдионе.

Во второй строке через пробел перечислены n различных чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$) — сами номиналы купюр.

Выходные данные

Выведите единственное число — минимальную *неудачную* сумму. Если же неудачных сумм не бывает, выведите -1.

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

E. Большой отрезок

2 секунды, 256 мегабайт

На координатной прямой задано n отрезков, i -ый отрезок начинается в позиции l_i и заканчивается в позиции r_i . Будем обозначать такой отрезок через $[l_i, r_i]$.

Вы предположили, что один из заданных отрезков покрывает все остальные. Другими словами, существует такой отрезок из заданного набора, в котором содержатся все остальные. Теперь вы хотите убедиться в своем предположении. Найдите отрезок из набора, который покрывает все остальные заданные отрезки, и выведите его номер. Если такого отрезка не существует, выведите -1.

Формально будем считать, что отрезок $[a, b]$ покрывает отрезок $[c, d]$, если выполняется условие $a \leq c \leq d \leq b$.

Входные данные

В первой строке задано целое число n ($1 \leq n \leq 10^5$) — количество отрезков. В следующих n строках содержатся описания отрезков. В i -ой строке через пробел заданы два целых числа l_i, r_i ($1 \leq l_i \leq r_i \leq 10^9$) — границы i -ого отрезка.

Гарантируется, что никакие два отрезка не совпадают.

Выходные данные

Выведите единственное целое число — номер отрезка, который покрывает все остальные отрезки набора. Если решения не существует, выведите -1.

Отрезки нумеруются с 1 в порядке их появления во входных данных.

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

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

F. Владик и сложная книга

2 секунды, 256 мегабайт

Владик начал читать сложно написанную книгу по алгоритмам, содержащую n страниц. Для того, чтобы улучшить понимание написанного, его друзья посоветовали ему читать страницы в некотором порядке, заданном перестановкой $P = [p_1, p_2, \dots, p_n]$, где p_i обозначает номер страницы, которую следует читать i -й по очереди.

Иногда мама Владика сортировала по возрастанию некоторый отрезок перестановки P с позиции l по r включительно, так как очень любит порядок. Для каждого из таких случаев Владик знает x — какую по счету страницу из перестановки он сейчас должен прочитать. Ему интересно, поменялась ли страница, которую он будет читать после сортировки, другими словами, поменялось ли p_x . После каждой такой сортировки он возвращал перестановку в изначальное состояние, поэтому можно считать, что каждая такая сортировка независима от других.

Входные данные

В первой строке находится два целых числа n, m ($1 \leq n, m \leq 10^4$) — длина перестановки и количество случаев.

В следующей строке входных данных задано n целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — элементы перестановки P . Напомним, что все элементы перестановки различны.

В каждой из m следующих строк задано по три целых числа l_i, r_i, x_i ($1 \leq l_i \leq x_i \leq r_i \leq n$) — левая и правая границы сортируемого подотрезка в i -м случае и позиция, которая интересует Владика, после сортировки.

Выходные данные

Для каждого случая в отдельной строке выведите «Yes», если страница, которую будет читать Владик, останется такой же и «No» иначе.

входные данные
5 5 5 4 3 2 1 1 5 3 1 3 1 2 4 3 4 4 4 2 5 3
выходные данные
Yes No Yes Yes No

входные данные
6 5 1 4 3 2 5 6 2 4 3 1 6 2 4 5 4 1 3 3 2 6 3
выходные данные
Yes No Yes No Yes

Объяснение первого примера:

- 1. [1, 2, 3, 4, 5] — перестановка после сортировки, так как 3-й элемент не изменился, ответ «Yes».
- 2. [3, 4, 5, 2, 1] — перестановка после сортировки, так как 1-й элемент изменился, ответ «No».
- 3. [5, 2, 3, 4, 1] — перестановка после сортировки, так как 3-й элемент не изменился, ответ «Yes».
- 4. [5, 4, 3, 2, 1] — перестановка после сортировки, так как 4-й элемент не изменился, ответ «Yes».
- 5. [5, 1, 2, 3, 4] — перестановка после сортировки, так как 3-й элемент изменился, ответ «No».

G. 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 выбираются 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 $a_1, a_2, ..., a_n$ и $b_1, b_2, ..., b_n$ 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.

H. 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 клеток, при этом для всех i ($1 < i \leq n$) выполняется $c_i \leq c_{i-1}$.

Обозначим за s общее количество клеток 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:

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