

Задача А. Хогвартс

Имя входного файла: `hogwarts.in`
Имя выходного файла: `hogwarts.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 32 мегабайта

Фродо - хоббит, привыкший к тихой размеренной жизни. Будучи самым сильным и умным в деревне, он жил в гармонии с природой и в хороших отношениях с другими хоббитами. Но все кардинально изменилось, когда напал народ огня... Фродо был единственным, кто мог их остановить. Но в тот момент, когда деревня нуждалась в нем как никогда, произошло что-то невообразимое. Дарт Ходор, предводитель народа огня, признался, что он отец Фродо. Эта новость оказалась настолько неожиданной для Фродо, что он потерял сознание. И пока Фродо был без сознания, пытаясь совладать с жестокой реальностью, он думал над одной задачей:

Дана перестановка чисел от 1 до N и две допустимые операции (описанные ниже), найдите наименьшее количество операций, необходимых для сортировки перестановки в возрастающем порядке.

Описание допустимых операций:

1. Поменять местами два соседних элемента перестановки
2. Поставить первый элемент в конец перестановки

И еще: после того, как Фродо воспользовался хотя бы одной операцией типа 1, он не может больше воспользоваться операцией типа 2.

Фродо верит, что решение данной задачи поможет ему вернуться в сознание и решить свои проблемы. Помогите Фродо с решением этой задачи.

Формат входных данных

В первой строке содержится целое число N ($1 \leq N \leq 10^6$). Во второй строке записаны N чисел, описывающих перестановку.

Формат выходных данных

Выведите в единственной строке целое число - ответ на задачу.

Примеры

hogwarts.in	hogwarts.out
5 5 1 2 4 3	2
6 1 6 2 3 4 5	4

Замечание

Оптимальное решение для первого примера - поставить 5 в конец перестановки, получив 1 2 4 3 5, а затем поменять местами 4 и 3: получив 1 2 3 4 5. Оптимальное решение для второго примера - поменять местами соседние элементы 4 раза: (6,2), (6,3), (6,4) и (6,5).

Задача В. IP адреса возвращаются!

Имя входного файла: `subnets.in`
Имя выходного файла: `subnets.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Интернет провайдеры (ISP) обычно владеют множеством IP адресов. С помощью них они предоставляют услуги своим клиентам. Одним из важных аспектов в управлении ISP является разделение адресов на подсети - непересекающиеся множества IP адресов. Для простоты будем считать, что следующие свойства выполняются для подсетей:

- Каждый IP адрес принадлежит ровно одной подсети;
- Каждая подсеть может содержать только адреса, принадлежащие Интернет провайдеру;
- Каждая подсеть определяется наидлиннейшим совпадающим префиксом¹ содержащихся в данной подсети IP адресов, представленных в двоичной записи. Этом префикс называется идентификатором подсети;
- Если в наличии у провайдера имеется подсеть с определенным идентификатором, то любой адрес, подходящий под префикс подсети должен принадлежать подсети.

Вам дан список всех IP адресов, принадлежащих провайдеру. Ваша задача определить наименьшее количество подсетей, на которое можно разбить все данные адреса.

Формат входных данных

Первая строка входного файла содержит число N ($1 \leq N \leq 100000$) — количество IP адресов во владении провайдера. Следующие N строк описывают IP адреса (версии IPv4) в точечно-числовом представлении A.B.C.D, где A, B, C и D — 8-битные целые числа, заданные в десятичной записи. Гарантируется, что все IP адреса в списке уникальны.

Формат выходных данных

Выведите единственное число в выходной файл — минимальное количество подсетей, на которые данные IP адреса могут быть разбиты.

Примеры

subnets.in	subnets.out
2 192.168.8.0 192.168.0.0	2
5 192.168.0.0 192.168.0.1 192.168.0.3 192.168.0.2 192.168.0.4	2

Замечание

¹Наидлиннейший совпадающий префикс IP адресов определяется следующим образом. Мы записываем IP адреса в двоичной форме и сравниваем их префиксы. Например, 192.168.0.1 и 192.168.8.0 имеют следующую двоичную запись:

11000000101010000000000000000001

11000000101010000000100000000000

Легко видеть, что их наидлиннейший общий префикс

11000000101010000000.

Задача С. Эксперимент

Имя входного файла: `marijuana.in`
Имя выходного файла: `marijuana.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

После переезда в Астану, Фуад бей решил начать новый бизнес - выращивать марихуану (конечно для медицинского применения) в своем загородном доме под Астаной. В начале года он купил семена P различных сортов марихуаны, чтобы посадить их и выяснить, какой вид лучше подходит для сурового Астанинского климата. Сорты пронумерованы числами от 1 до P . Позже выяснилось, что плодородие почвы на участке довольно неравномерно, поэтому он решил посадить семена следующим образом. Для начала он разделил свое квадратное поле на $P \times P$ маленьких одинаковых квадратных клеток, которые образуют поле с P рядами и P столбцами. Далее он собирается садить семена, так что в каждую клетку попадет ровно один семя. Он также хочет распределить семена одного сорта более или менее равномерно по всему полю, так что он рассчитывает, чтобы каждый вид марихуаны будет присутствовать в каждом из P рядов и в каждом из P столбцов. Таким образом, он собирается посадить P^2 семян, по одной в каждой ячейке.

В дополнение к семенам Фуад бей приобрел P различных типов удобрений, которыми он собирается удобрять поле. Они пронумерованы числами от 1 до P . Проблема в том, что он не знает влияние почвы на эффективность удобрений, и что еще важнее, он не может предсказать, как каждый тип удобрения будет влиять на рост каждого сорта марихуаны. Поэтому он придумал способ, как применить удобрения к семенам. Во-первых, к каждому семени может быть применен ровно один тип удобрений. Во-вторых, в каждом ряду и в каждом столбце поля все P типов удобрений должны присутствовать. Наконец, каждый тип удобрения должен быть использовать с каждым сортом марихуаны, что поможет ему определить наилучшее сочетание удобрения и сорта марихуаны. У Фуад бея нет времени на проектирование схемы для посадки и распределения удобрений, поэтому он предложил эту задачу Вам. Он специально выбрал P простым, так как считает, что это упростит Вам задачу.

Формат входных данных

Первая строка входного файла содержит простое целое P ($2 \leq P \leq 1009$) — количество сортов марихуаны и число видов удобрений, которые купил Фуад бей.

Формат выходных данных

Если нет способа посеять семена и удобрить их как хочет Фуад бей, выведите -1. В противном случае, выведите $2P$ строк, каждая содержащая P целых числа. Первые P строк должны содержать описание того, как Фуад бей должен посадить семена марихуаны. j -й число в i -ой строке должно содержать целое число от 1 и P включительно, описывающее сорт марихуаны, которое Фуад бей должен посадить в i -ом ряду j -го столбца участка. Следующие P строк должны описывать как должны быть распределены P типов удобрений в том же формате. Типы удобрений должны быть пронумерованы числами от 1 до P . Если имеется несколько решений, Вы можете вывести любое из них.

Примеры

<code>marijuana.in</code>	<code>marijuana.out</code>
2	-1
3	1 2 3 3 1 2 2 3 1 3 1 2 1 2 3 2 3 1

Замечание

Во втором примере, если мы сопоставим сорт семян с соответствующим типом удобрений, мы

получим пары

(1, 3), (2, 1), (3, 2)

(3, 1), (1, 2), (2, 3)

(2, 2), (3, 3), (1, 1)

Легко проверить, что эти девять пар чисел уникальны — каждый вид марихуаны встречается с каждым типом удобрения ровно один раз.

Задача D. Делимые пары

Имя входного файла: `pairs.in`
Имя выходного файла: `pairs.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дана перестановка из n чисел a_1, a_2, \dots, a_n . Более того, заданы m запросов, каждый из которых характеризуется двумя целыми числами l_i и r_i . Ответом на запрос является количество пар целых чисел p и q , таких что $l_i \leq p < q \leq r_i$ и либо a_p делится на a_q либо a_q делится на a_p . Вашей задачей является найти ответ для каждого из заданных запросов.

Формат входных данных

В первой строке задано одно целое число n ($1 \leq n \leq 3 \cdot 10^5$). Во второй строке заданы n различных целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$) – элементы перестановки. В третьей строке задано одно целое число m ($1 \leq m \leq 5 \cdot 10^5$) – количество запросов. В каждой из следующих m строк заданы по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq n$) описывающие очередной запрос.

Формат выходных данных

Для каждого запроса выведите одно целое число в отдельной строке — ответ на запрос.

Примеры

<code>pairs.in</code>	<code>pairs.out</code>
8	3
7 1 5 6 2 3 4 8	3
5	2
1 4	6
5 8	1
4 6	
2 6	
3 5	

Задача Е. Максимальный прямоугольник

Имя входного файла: `rectangle.in`
Имя выходного файла: `rectangle.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам дана матрица из n строк и m столбцов, заполненная целыми положительными числами, не превосходящими 10^9 . Напишите программу, которая найдет в ней площадь наибольшего прямоугольника, такого, что в нем нет двух чисел, разность между которыми превосходила бы T .

Формат входных данных

В первой строке входных данных даны три числа, разделенных пробелами n , m , T ($1 \leq n \leq 300$, $1 \leq m \leq 300$, $0 \leq T \leq 10^9$). В следующих n строках даны по m чисел — элементы матрицы.

Формат выходных данных

Выведите одно число - ответ на задачу

Примеры

rectangle.in	rectangle.out
4 3 5 1 10 7 5 4 8 1 3 7 100 6 6	6
1 8 0 1 2 3 3 4 5 5 5	3

Задача F. Heavy Light декомпозиция

Имя входного файла: hld.in
Имя выходного файла: hld.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Все вы наверное слышали про heavy-light декомпозицию? Чтобы получить heavy-light декомпозицию корневого дерева, нужно каждое ребро назвать легким, либо тяжелым. Выбор зависит от того, выполняется ли следующее условие:

Допустим $s(v)$ — количество вершин в поддереве вершины v . Тогда ребро $u - v$, где u предок v , будет тяжелым, если $\lceil \frac{s(u)}{2} \rceil \leq s(v)$, иначе — легким.

Ваша задача очень проста: посчитать количество корневых неизоморфных деревьев из n вершин, таких что их heavy-light декомпозиция содержит ровно k тяжелых ребер.

Два дерева называются изоморфными, если можно перенумеровать вершины одного дерева так, чтобы получить другое дерево.

Формат входных данных

В первой строке входных данных содержится одно целое число T ($1 \leq T \leq 820$) — количество тестов.

В следующих T строках содержится по два целых числа разделенных пробелом k и n ($0 \leq k < n \leq 40$).

Формат выходных данных

Выведите T строк, каждая из которых содержит ответ на соответствующий тест.

Примеры

hld.in	hld.out
3	0
3 1	1
3 0	1
1 0	

Задача G. XOR XOR XOR

Имя входного файла: `xor.in`
Имя выходного файла: `xor.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Задано множество целых чисел $\{1, 2, \dots, n\}$, посчитайте количество его непустых подмножеств, XOR чисел которых равен x .

Формат входных данных

В первой строке входных данных задано число T ($1 \leq T \leq 10^5$) — количество тестов. В следующих T строках заданы по два целых числа n и x ($1 \leq n \leq 10^{18}$, $0 \leq x \leq 10^{18}$), разделенных одним пробелом.

Формат выходных данных

Выведите T чисел, по одному в каждой строке, ответы на соответствующие тесты.

Примеры

<code>xor.in</code>	<code>xor.out</code>
2	1
2 2	0
3 4	