

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

A. Любовный треугольник

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Как известно, нет самолетов-мужчин и самолетов-женщин. Однако, каждому самолету на Земле нравится какой-то один другой самолет. Всего на Земле n самолетов, пронумерованных от 1 до n , при этом самолету номер i нравится самолет f_i , где $1 \leq f_i \leq n$, а также $f_i \neq i$.

Назовем любовным треугольником ситуацию, когда самолету A нравится самолет B , самолету B нравится самолет C , а самолету C нравится самолет A . Проверьте, есть ли на Земле хотя бы один любовный треугольник.

Входные данные

Первая строка содержит одно целое число n ($2 \leq n \leq 5000$) — число самолетов.

Вторая строка содержит n целых чисел f_1, f_2, \dots, f_n ($1 \leq f_i \leq n, f_i \neq i$), что означает, что i -му самолету нравится f_i -й.

Выходные данные

Выведите «YES», если на Земле есть любовный треугольник из самолетов. Иначе выведите «NO».

Каждую букву можно выводить в любом регистре: строчную или заглавную.

Примеры

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

Примечание

В первом примере самолету 2 нравится самолет 4, самолету 4 нравится самолет 1, а самолету 1 нравится самолет 2, поэтому это любовный треугольник.

Во втором примере любовных треугольников нет.



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

A. Праздник

ограничение по времени на тест: 3 seconds

ограничение по памяти на тест: 256 megabytes

В компании работает n сотрудников, пронумерованных от 1 до n . У каждого сотрудника либо нет руководителя, либо есть ровно один непосредственный руководитель — некоторый другой сотрудник с другим номером. Сотрудник A называется начальником другого сотрудника B , если выполняется хотя бы одно из двух условий:

- Сотрудник A — непосредственный руководитель сотрудника B .
- У сотрудника B есть непосредственный руководитель, сотрудник C , такой, что A является начальником сотрудника C .

В структуре компании нет циклов. То есть никакой сотрудник не является начальником своего непосредственного руководителя.

Сегодня компания собирается организовать праздник. Для этого необходимо разделить всех n сотрудников на несколько групп: каждый человек должен относиться ровно к одной группе. Более того, в каждой группе не должно быть таких двух сотрудников A и B , что A является начальником B .

Ваша задача — найти наименьшее возможное количество таких групп.

Входные данные

Первая строка содержит целое число n ($1 \leq n \leq 2000$) — количество сотрудников.

Следующие n строк содержат целые числа p_i ($1 \leq p_i \leq n$ или $p_i = -1$). Каждое p_i обозначает непосредственного руководителя i -го сотрудника. Если p_i равно -1 , то i -й сотрудник не имеет непосредственного руководителя.

Гарантируется, что никакой сотрудник не является своим собственным непосредственным руководителем ($p_i \neq i$). Также гарантируется, что структура компании не содержит циклов.

Выходные данные

Выведите единственное целое число — минимальное количество групп, на которые можно разделить всех сотрудников.

Примеры

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

Примечание

В первом примере достаточно трех групп:

- Сотрудник 1
- Сотрудники 2 и 4
- Сотрудники 3 и 5




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

A. Транспорт на Новый год

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Новый год приходит в Линейный мир! В этом мире есть n ячеек, пронумерованных целыми числами от 1 до n , уложенных в виде доски размером $1 \times n$. В ячейках живут люди. Однако, передвигаться между различными ячейками сложно, ведь выйти из ячейки — дело непростое. В то же время, люди хотят знакомиться с людьми, живущими в других ячейках.

И вот, tncks0121 придумал систему транспорта для передвижения между ячейками, чтобы люди могли отпраздновать Новый год. Сперва он задумал $n - 1$ положительных целых чисел a_1, a_2, \dots, a_{n-1} . Для каждого целого числа i , где $1 \leq i \leq n - 1$, выполняется условие $1 \leq a_i \leq n - i$. Затем он создал $n - 1$ порталов, пронумерованных целыми числами от 1 до $n - 1$. Из них i -й ($1 \leq i \leq n - 1$) портал соединяет ячейку номер i и ячейку номер $(i + a_i)$, т. е. с его помощью можно путешествовать из ячейки i в ячейку $(i + a_i)$. К сожалению, портал не работает в обратную сторону, то есть нельзя пройти из ячейки $(i + a_i)$ в ячейку i по i -му порталу. Легко заметить, что из-за условия $1 \leq a_i \leq n - i$ нельзя покинуть Линейный мир, пользуясь порталами.

Я нахожусь в ячейке 1 и хочу пройти в ячейку t . Однако я не знаю, могу ли я там оказаться. Пожалуйста, определите, могу ли я пройти в ячейку t , пользуясь только построенной системой транспорта.

Входные данные

В первой строке записано два целых числа через пробел, n ($3 \leq n \leq 3 \times 10^4$) и t ($2 \leq t \leq n$) — количество ячеек и номер ячейки, в которую я хочу попасть.

Во второй строке записано $n - 1$ целых чисел через пробел a_1, a_2, \dots, a_{n-1} ($1 \leq a_i \leq n - i$). Гарантируется, что пользуясь данной транспортной системой, покинуть Линейный мир нельзя.

Выходные данные

Если я могу дойти до ячейки t по данной транспортной системе, выведите "YES". В противном случае, выведите "NO".

Примеры

входные данные	Скопировать
8 4 1 2 1 2 1 2 1	
выходные данные	Скопировать
YES	

входные данные	Скопировать
8 5 1 2 1 2 1 1 1	
выходные данные	Скопировать
NO	

Примечание

В первом примере можно дойти до t , посетив следующие ячейки: 1, 2, 4.

Во втором примере можно посетить лишь ячейки 1, 2, 4, 6, 7, 8; значит, мы не можем попасть в требуемую ячейку 5.




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

C. Длинные прыжки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарп нашёл под ёлкой массив a из n элементов и инструкцию для игры с ним:

- Выбери число i ($1 \leq i \leq n$) — стартовую позицию в массиве. Помести фишку на позицию i (на элемент a_i);
- Пока $i \leq n$, прибавь к своему результату значение a_i и переместись на a_i вправо (т.е. замени i на $i + a_i$);
- Если $i > n$, то Поликарп заканчивает игру.

Например, если $n = 5$ и $a = [7, 3, 1, 2, 3]$, тогда следующие варианты игр возможны:

- Поликарп выбирает $i = 1$. Процесс игры: $i = 1 \xrightarrow{+7} 8$. Результат игры: $a_1 = 7$.
- Поликарп выбирает $i = 2$. Процесс игры: $i = 2 \xrightarrow{+3} 5 \xrightarrow{+3} 8$. Результат игры: $a_2 + a_5 = 6$.
- Поликарп выбирает $i = 3$. Процесс игры: $i = 3 \xrightarrow{+1} 4 \xrightarrow{+2} 6$. Результат игры: $a_3 + a_4 = 3$.
- Поликарп выбирает $i = 4$. Процесс игры: $i = 4 \xrightarrow{+2} 6$. Результат игры: $a_4 = 2$.
- Поликарп выбирает $i = 5$. Процесс игры: $i = 5 \xrightarrow{+3} 8$. Результат игры: $a_5 = 3$.

Помогите Поликарпу узнать, какой максимальный результат он может получить, если он выбирает стартовую позицию оптимально.

Входные данные

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

В первой строке каждого набора входных данных находится одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — длина массива a .

В следующей строке находятся 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 в отдельной строке выведите одно число — максимальный результат, который Поликарп может получить, сыграв в игру на соответствующем массиве по инструкции из условия. Обратите внимание, что Поликарп выбирает любую стартовую позицию от 1 до n таким образом, чтобы максимизировать свой результат.

Пример

входные данные	Скопировать
<pre>4 5 7 3 1 2 3 3 2 1 4 6 2 1000 2 3 995 1 5 1 1 1 1 1</pre>	

выходные данные	Скопировать
<pre>7 6 1000 5</pre>	

Примечание

Первый набор входных данных разобран в условии.

Во втором наборе входных данных максимальный результат достигается при выборе $i = 1$.

В третьем наборе входных данных максимальный результат достигается при выборе $i = 2$.

В четвёртом наборе входных данных максимальный результат достигается при выборе $i = 1$.


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

E. Озера

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Дана сетка a размера $n \times m$ из неотрицательных целых чисел, где $a_{i,j}$ представляет глубину воды в i -й строке и j -м столбце.

Озеро — это набор ячеек, таких что:

- Каждая ячейка в наборе имеет $a_{i,j} > 0$
- Существует путь между любой парой ячеек в озере, двигаясь вверх, вниз, влево или вправо, несколько раз и не наступая на ячейку с $a_{i,j} = 0$.

Объем озера — это сумма глубин всех ячеек в озере.

Найдите наибольший объем озера в сетке.

Входные данные

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

Первая строка каждого набора содержит два целых числа n, m ($1 \leq n, m \leq 1000$) — размеры сетки.

Затем следуют n строк, каждая из которых содержит m целых чисел $a_{i,j}$ ($0 \leq a_{i,j} \leq 1000$) — глубина воды в каждой ячейке.

Гарантируется, что сумма $n \cdot m$ по всем наборам не превышает 10^6 .

Выходные данные

Для каждого теста выведите одно целое число — наибольший объем озера в сетке.

Пример

входные данные	Скопировать
<pre>5 3 3 1 2 0 3 4 0 0 0 5 1 1 0 3 3 0 1 1 1 0 1 1 1 1 5 5 1 1 1 1 1 0 0 0 1 1 0 5 0 1 1 0 0 0 1 1 1 1 1 1 5 5 1 1 1 1 1 1 0 0 0 1 1 1 4 0 1 1 0 0 0 1 1 1 1 1 1</pre>	
выходные данные	Скопировать
<pre>10 0 7 16 21</pre>	

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

A. Катание на Коньках

ограничение по времени на тест: 2 seconds

ограничение по памяти на тест: 256 megabytes

Байтек учится кататься на льду. Он новичок, и поэтому он передвигается следующим образом: сначала отталкивается от сугроба на север, восток, юг или запад — и катится до тех пор, пока не повстречает очередной сугроб. Байтек заметил, что таким образом он не сможет добраться от некоторых сугробов до некоторых других, какой бы ни была последовательность его движений. Теперь он хочет соорудить несколько дополнительных сугробов так, чтобы он мог добраться от любого сугроба до любого другого. Байтек попросил Вас найти наименьшее количество сугробов, которые ему потребуется соорудить.

Мы предполагаем, что Байтек может сооружать сугробы только в точках с целочисленными координатами.

Входные данные

В первой строке входного файла записано единственное целое число n ($1 \leq n \leq 100$) — количество сугробов. Каждая из следующих n строк содержит по два целых числа x_i и y_i ($1 \leq x_i, y_i \leq 1000$) — координаты i -ого сугроба.

Обратите внимание, что направление на север совпадает с направлением оси Oy , таким образом, направление на восток совпадает с направлением оси Ox . Все сугробы расположены в различных точках.

Выходные данные

Выведите наименьшее количество сугробов, которые надо соорудить Байтку для того, чтобы он мог добраться от любого сугроба до любого другого.

Примеры

входные данные	выходные данные
2 2 1 1 2	<button>Скопировать</button>
1	<button>Скопировать</button>
входные данные	выходные данные
2 2 1 4 1	<button>Скопировать</button>
0	<button>Скопировать</button>




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

C. Mocha и прогулка

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Mocha живет в городском уезде Чжицзян. В уезде $n + 1$ поселение и $2n - 1$ ориентированная дорога.

Все дороги имеют один из двух типов:

- $n - 1$ дорога идет от поселения i к поселению $i + 1$, для всех $1 \leq i \leq n - 1$.
- n дорог описываются последовательностью a_1, \dots, a_n . Если $a_i = 0$, то i -я из этих дорог идет от поселения i к поселению $n + 1$, иначе она идет от поселения $n + 1$ к поселению i , для всех $1 \leq i \leq n$.

Mocha собирается в поход вместе с Taki. Чтобы поход не был скучным, они собираются посетить каждое поселение **ровно один раз**. Они могут начать и закончить в любых поселениях. Можете помочь составить им план похода?

Входные данные

Во входных данных находятся несколько наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 20$) — количество наборов входных данных. Далее следуют наборы входных данных, каждый в двух строках.

Первая строка набора входных данных содержит одно целое число n ($1 \leq n \leq 10^4$), что означает, что всего поселений $n + 1$.

Вторая строка содержит n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 1$). Если $a_i = 0$, то есть дорога из поселения i в поселение $n + 1$. Если $a_i = 1$, то есть дорога из поселения $n + 1$ в поселение i .

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 10^4 .

Выходные данные

Для каждого набора входных данных выведите одну строку с $n + 1$ целыми числами, где i -е число — поселение, которое они должны посетить i -м по порядку. Если ответа не существует, выведите -1 .

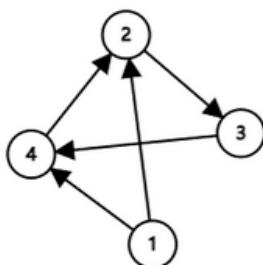
Если существует несколько решений, выведите любое из них.

Пример

входные данные	Скопировать
2	
3	
0 1 0	
3	
1 1 0	
выходные данные	Скопировать
1 4 2 3	
4 1 2 3	

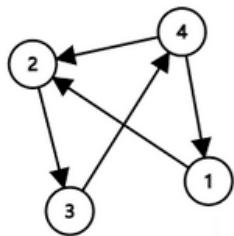
Примечание

В первом наборе входных данных уезд выглядит следующим образом:



Возможные ответы: $(1 \rightarrow 4 \rightarrow 2 \rightarrow 3)$ или $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4)$.

Во втором наборе входных данных уезд выглядит следующим образом:



Возможные ответы — $(4 \rightarrow 1 \rightarrow 2 \rightarrow 3)$, $(1 \rightarrow 2 \rightarrow 3 \rightarrow 4)$, $(3 \rightarrow 4 \rightarrow 1 \rightarrow 2)$ и $(2 \rightarrow 3 \rightarrow 4 \rightarrow 1)$.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:36:52^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



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

C. Слух

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вова пообещал себе больше никогда не играть в игры... Но недавно компания Firestorm выпустила очень популярную игру — World of Farcraft, и Вова, конечно же, начал в неё играть.

Он впал в ступор при выполнении очередного квеста — необходимо было прийти в поселение Надгород и распространить там слух.

Вова знает, что в Надгороде живут n персонажей. Некоторые персонажи дружат между собой и готовы делиться информацией друг с другом. Также он знает, что i -й персонаж готов начать распространять слух за c_i золота. Как только персонаж узнаёт слух, он рассказывает его всем своим друзьям, которые, в свою очередь, тоже начинают распространять этот слух (но уже бесплатно).

Вова хочет, чтобы слух узнали все n персонажей Надгорода. За какое минимальное количество золота он сможет это сделать?

Посмотрите примечание для лучшего понимания задачи.

Входные данные

В первой строке входных данных задано два целых числа n и m ($1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$) — количество людей в Надгороде и количество дружеских связей между ними.

В следующей строке входных данных заданы n целых чисел c_i ($0 \leq c_i \leq 10^9$) — количество золота, за которое i -й персонаж готов начать распространять слух.

В следующих m строках заданы пары (x_i, y_i) , обозначающие, что персонажи x_i и y_i дружат ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$). Гарантируется, что никакая пара дружащих персонажей не повторяется.

Выходные данные

Выведите единственное число — минимальное количество золота, за которое Вова сможет выполнить квест.

Примеры

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

входные данные	Скопировать
10 0 1 2 3 4 5 6 7 8 9 10	
выходные данные	Скопировать
55	

входные данные	Скопировать
10 5 1 6 2 7 3 8 4 9 5 10 1 2 3 4 5 6 7 8 9 10	
выходные данные	Скопировать
15	

Примечание

В первом тестовом примере Вове выгоднее всего заплатить первому персонажу (он узнает слух и расскажет его четвёртому персонажу, который в свою очередь расскажет его пятому персонажу). Также ему будет необходимо заплатить второму и

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

B. Махмуд, Ехаб и двудольность

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Махмуд и Ехаб продолжают свои приключения! Каждый житель Злой Страны знает, что Доктор Зло любит двудольные графы, особенно деревья.

Дерево — это связный граф без циклов. Двудольный граф — это граф, вершины которого можно разбить на 2 множества таким образом, что для любого ребра (u, v) графа вершины u и v лежат в разных множествах. Более формальное определение дерева и двудольного графа дано ниже.

Доктор Зло дал Махмуду и Ехабу дерево, состоящее из n вершин и сказал добавлять рёбра таким образом, чтобы граф оставался двудольным, а также в нём не было петель и кратных рёбер. Какое максимальное число рёбер они могут добавить?

Входные данные

В первой строке дано целое число n — число вершин в дереве ($1 \leq n \leq 10^5$).

В следующих $n - 1$ строках содержатся пары целых чисел u и v ($1 \leq u, v \leq n$, $u \neq v$) — описание рёбер дерева.

Гарантируется, что заданный граф является деревом.

Выходные данные

Выведите одно число — максимальное число рёбер, которые Махмуд и Ехаб могут добавить в граф.

Примеры

входные данные	Скопировать
3 1 2 1 3	
выходные данные	Скопировать
0	
входные данные	Скопировать
5 1 2 2 3 3 4 4 5	
выходные данные	Скопировать
2	

Примечание

Определение дерева: [https://ru.wikipedia.org/wiki/Дерево_\(теория_графов\)](https://ru.wikipedia.org/wiki/Дерево_(теория_графов))

Определение двудольного графа: https://ru.wikipedia.org/wiki/Двудольный_граф

В первом тестовом примере единственное ребро, которое можно добавить в граф, чтобы избежать появления петель и кратных рёбер — это $(2, 3)$, но его добавление сделает граф не двудольным.

Во втором тестовом примере Махмуд и Ехаб могут добавить рёбра $(1, 4)$ и $(2, 5)$.


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

F. Вечная зима

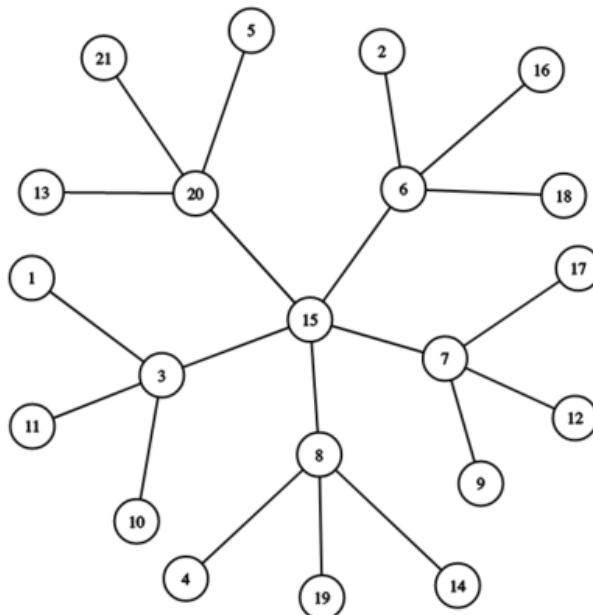
ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Граф-снежинка генерируется из двух целых чисел x и y , которые больше 1, следующим образом:

- Начните с одной центральной вершины.
- Подключите x новых вершин к этой центральной вершине.
- Подключите y новых вершин к **каждой** из этих x вершин.

Например, ниже приведен граф-снежинка для $x = 5$ и $y = 3$.



Граф-снежинка выше имеет центральную вершину 15, затем $x = 5$ вершин, подключенных к ней (3, 6, 7, 8 и 20), а затем $y = 3$ вершины, подключенные к каждой из них.

Для заданного графа-снежинки определите значения x и y .

Входные данные

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

Первая строка каждого набора входных данных содержит два целых числа n и m ($2 \leq n \leq 200$;

$1 \leq m \leq \min\left(1000, \frac{n(n-1)}{2}\right)$) — количество вершин и ребер в графе соответственно.

Следующие m строк содержат по два целых числа u и v ($1 \leq u, v \leq n$, $u \neq v$) — номера вершин, соединенных ребром. Граф не содержит кратных ребер и петель.

Гарантируется, что этот граф является графом снежинки для некоторых целых чисел x и y , которые больше 1.

Выходные данные

Для каждого набора входных данных на отдельной строке выведите значения x и y , в этом порядке, разделенные пробелом.

Пример

входные данные	Скопировать
<pre>3 21 20 21 20 5 20 13 20 1 3 11 3 10 3 4 8 19 8</pre>	

```
14 8
9 7
12 7
17 7
18 6
16 6
2 6
6 15
7 15
8 15
20 15
3 15
7 6
1 2
1 3
2 4
2 5
3 6
3 7
9 8
9 3
3 6
6 2
2 1
5 2
2 7
4 3
3 8
```

ВХОДНЫЕ ДАННЫЕ**Скопировать**

```
5 3
2 2
2 3
```

Примечание

Первый набор входных данных изображен в условии. Обратите внимание, что вывод 3 5 является **неправильным**, так как сначала должно быть выведено x , а затем y .

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:37:07^{UTC+5} (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО


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

B. Строим дорогу

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В одной стране есть n городов. Изначально в этой стране нет дорог. Но вот, однажды король решает построить дороги, соединяющие пары городов. По дорогам можно путешествовать в любую сторону. Король хочет построить дороги так, чтобы можно было добраться из любого города в любой другой, проехав не более двух дорог. Также дано m пар городов — между этими парами городов дороги строить нельзя.

Ваша задача — построить как можно меньше дорог так, чтобы вышеперечисленные условия выполнялись. Ограничения гарантируют, что это возможно всегда.

Входные данные

Первая строка содержит два целых числа, n и m ($1 \leq n \leq 10^3$, $0 \leq m < \frac{n}{2}$).

Затем следует m строк, каждая строка состоит из двух целых чисел, a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), показывающих, что нельзя строить дорогу между городами a_i и b_i . Считайте, что города пронумерованы от 1 до n .

Гарантируется, что каждая пара городов упоминается во входных данных ровно один раз.

Выходные данные

В первой строке выведите целое число s : наименьшее количество дорог, которые надо построить. Затем следует вывести s строк, по два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) в каждой, означающих, что надо построить дорогу между городами a_i и b_i .

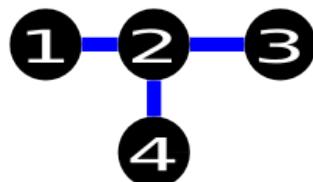
Если существует несколько решений, можно выводить любое.

Примеры

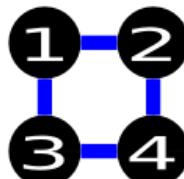
входные данные	Скопировать
4 1 1 3	
выходные данные	Скопировать
3 1 2 4 2 2 3	

Примечание

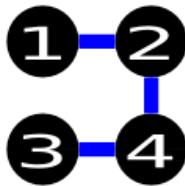
Ниже показано одно из возможных решений тестового примера:



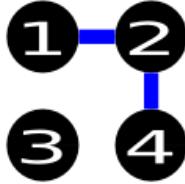
Примеры неправильных решений:



Решение, изображенное на картинке выше, неверно из-за того, что количество построенных дорог не минимально (4 против 3). Также, здесь строится дорога между городами 1 и 3, несмотря на то, что во входных данных указан запрет на строительство дороги между этой парой.



Решение, изображенное на картинке выше, неверно, так как нужно проехать по меньшей мере 3 дороги, чтобы попасть из города 1 в город 3, а в Вашем ответе должно быть возможно добраться из любого города в любой другой, пройдя не более 2 дорог.



И наконец, это решение неверно потому, что должна быть возможность добраться от любого города до любого другого, а в этом ответе невозможно добраться от 1 до 3, от 2 до 3, и от 4 до 3.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:37:08^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



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

B. Две кнопки

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вася нашел странное устройство. На передней панели устройства расположены дисплей, отображающий некоторое целое положительное число, а также красная и синяя кнопки. При нажатии на красную кнопку число на дисплее увеличивается в два раза. При нажатии на синюю кнопку число на дисплее уменьшается на единицу. Если в какой-то момент число перестает быть положительным, устройство ломается. Дисплей может отображать сколь угодно большие числа. Изначально на дисплее отображается число n .

Вася хочет получить на дисплее число m . Какое минимальное количество нажатий ему придется совершить?

Входные данные

Первая и единственная строка ввода содержит два различных целых числа n и m , разделенных пробелом ($1 \leq n, m \leq 10^4$).

Выходные данные

Выведите одно число — минимальное количество нажатий на кнопки, необходимое, чтобы получить число m из числа n .

Примеры

входные данные	Скопировать
4 6	
выходные данные	Скопировать
2	
входные данные	Скопировать
10 1	
выходные данные	Скопировать
9	

Примечание

В первом примере нужно один раз нажать на синюю кнопку, и затем раз один раз на красную кнопку.

Во втором примере удваивать число невыгодно, поэтому надо девять раз нажать на синюю кнопку.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:37:16^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО


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

C. Бал в Берляндии

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В школе, в которой учится Вася, идет подготовка к проведению выпускного. Одно из запланированных выступлений — бал, в котором примут участие пары состоящие из мальчиков и девочек.

Каждый класс должен представить на бал по две пары. В классе Васи a мальчиков и b девочек изъявили желание участвовать в мероприятии. Но не все мальчики и не все девочки готовы танцевать в паре.

Формально, вам известно k возможных пар, состоящих из одного мальчика и одной девочки. Вам нужно выбрать из этих пар две так, чтобы ни один человек не состоял больше чем в одной паре.

Например, если $a = 3$, $b = 4$, $k = 4$ и следующие пары готовы танцевать вместе $(1, 2)$, $(1, 3)$, $(2, 2)$, $(3, 4)$ (в каждой паре сначала идет номер мальчика, потом номер девочки), то возможны следующие комбинации из двух пар (ниже перечислены не все возможные варианты):

- $(1, 3)$ и $(2, 2)$;
- $(3, 4)$ и $(1, 3)$;

Но следующие комбинации не возможны:

- $(1, 3)$ и $(1, 2)$ — первый мальчик входит сразу в две пары;
- $(1, 2)$ и $(2, 2)$ — вторая девочка входит сразу в две пары;

Найдите количество способов выбрать две пары, походящие под условие выше. Два способа считаются различными, если состоят из разных пар.

Входные данные

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

В первой строке каждого набора входных данных содержатся три целых числа a , b и k ($1 \leq a, b, k \leq 2 \cdot 10^5$) — количество мальчиков и девочек в классе и количество пар готовых танцевать вместе.

В второй строке каждого набора входных данных содержится k целых чисел a_1, a_2, \dots, a_k . ($1 \leq a_i \leq a$), где a_i — номер мальчика в паре с номером i .

В третьей строке каждого набора входных данных содержится k целых чисел b_1, b_2, \dots, b_k . ($1 \leq b_i \leq b$), где b_i — номер девочки в паре с номером i .

Гарантируется, что суммы a , b и k по всем наборам входных данных не превосходят $2 \cdot 10^5$.

Гарантируется, что каждая пара указана в одном наборе входных данных не больше одного раза.

Выходные данные

Для каждого набора входных данных в отдельной строке выведите одно целое число — количество способов выбрать две пары, походящие под условие выше.

Пример

входные данные	Скопировать
<pre>3 3 4 4 1 1 2 3 2 3 2 4 1 1 1 1 1 2 2 4 1 1 2 2 1 2 1 2</pre>	
выходные данные	Скопировать
<pre>4 0 2</pre>	

Примечание

В первом наборе входных данных следующие комбинации из двух пар подходят:

- (1, 2) и (3, 4);
- (1, 3) и (2, 2);
- (1, 3) и (3, 4);
- (2, 2) и (3, 4).

Во втором наборе входных данных есть всего одна пара.

В третьем наборе входных данных следующие комбинации из двух пар подходят:

- (1, 1) и (2, 2);
- (1, 2) и (2, 1).

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:37:16^{UTC+5} (1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке




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

A. Паутина лжи

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Когда играешь в игру престолов, ты побеждаешь или умираешь. Третьего не дано.

Серсия Ланнистер, *Игра престолов*, Джордж Р. Р. Мартин

Есть n дворян, пронумерованных от 1 до n . Дворянин i обладает могуществом i . Дворяне «дружат» между собой — всего есть m пар дружных дворян. Дружба между дворянами a и b всегда взаимная.

Дворянин считается **уязвимым**, если выполнены оба следующих условия.

- у этого дворянина есть хотя бы один друг, и
- **все** друзья этого дворянина обладают могуществом, превосходящим могущество этого дворянина.

Вам нужно будет обрабатывать следующие три типа запросов.

1. Добавить дружбу между u и v .
2. Удалить дружбу между u и v .
3. Посчитать ответ на следующий процесс.

Процесс: все уязвимые дворяне одновременно умирают, и все их дружбы заканчиваются. После этого, возможно, некоторые другие дворяне становятся уязвимыми. Процесс повторяется до тех пор, пока есть уязвимые дворяне. Можно доказать, что процесс завершится за конечное число шагов. После завершения процесса вам необходимо посчитать количество оставшихся дворян.

Заметьте, что результаты процессов не влияют на дальнейшие и предыдущие запросы, то есть все дворяне живы в начале каждого процесса!

Входные данные

Первая строка входных данных содержит два целых числа n и m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$) — количество дворян и количество изначальных пар друзей соответственно.

Следующие m строк содержат по два целых числа u и v ($1 \leq u, v \leq n$, $u \neq v$), описывающие дружбу между дворянами. Никакая пара друзей не повторяется дважды.

Следующая строка содержит целое число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов.

Следующие q строк содержат сами запросы по одному в строке. Каждый запрос имеет один из следующих видов:

- 1 u v ($1 \leq u, v \leq n$, $u \neq v$) — добавить дружбу между u и v . Гарантируется, что u и v не друзья в этот момент.
- 2 u v ($1 \leq u, v \leq n$, $u \neq v$) — удалить дружбу между u и v . Гарантируется, что u и v дружат в этот момент.
- 3 — вывести ответ на процесс, описанный в условии.

Выходные данные

На каждый запрос 3-го типа выведите целое число, по одному на строку. Гарантируется, что будет хотя бы один запрос типа 3.

Примеры

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

входные данные**Скопировать**

```
4 3
2 3
3 4
4 1
1
3
```

выходные данные**Скопировать****1****Примечание**

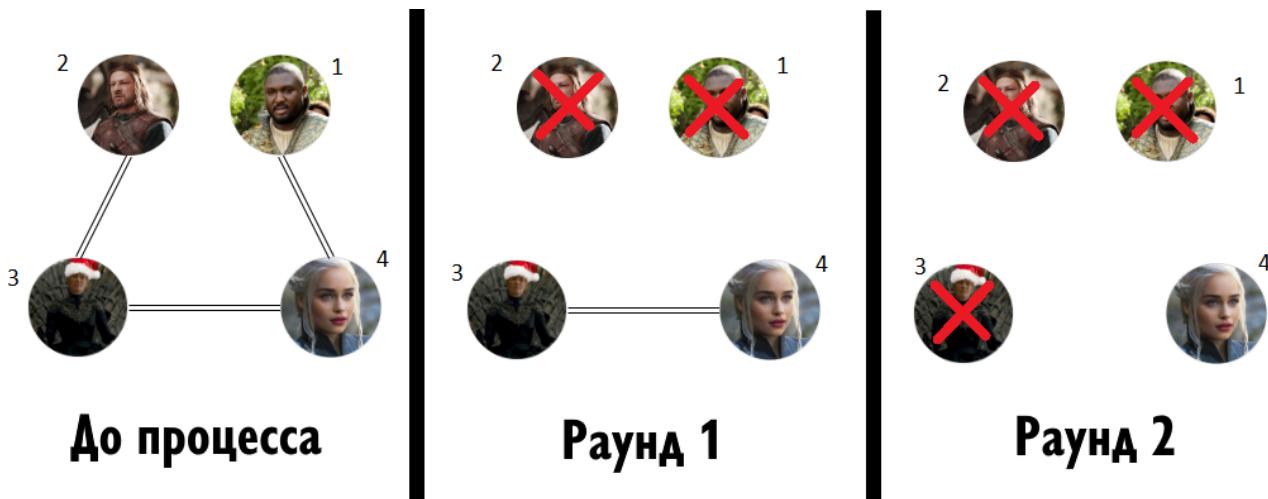
Ситуация в первом запросе типа 3 изображена на рисунке ниже.

В первом раунде процесса дворянин 1 слабее, чем все его друзья (2 и 3), и поэтому он умирает. Никакой другой дворянин в раунде 1 не является уязвимым. В раунде 2 дворянин 3 слабее своего единственного друга, дворянина 4, и поэтому он умирает. На этом этапе процесс завершается, ответ равен 2.



Во втором запросе типа 3 единственным выжившим является дворянин 4.

Второй пример состоит только из одного запроса типа 3. В первом раунде умирают два дворянина, во втором раунде умирает один дворянин. Итоговый ответ равен 1, поскольку только один дворянин выжил.




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

D. Продуктивная встреча

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На встрече собрались n человек. В каждый момент времени любые два человека могут отойти и поговорить наедине. Одни и те же два человека могут поговорить несколько (сколько угодно) раз за встречу.

У каждого человека есть ограниченная общительность. Общительность i -го человека равна неотрицательному целому числу a_i . Это означает, что после a_i личных разговоров этот человек уходит со встречи (и больше ни с кем не разговаривает). Если $a_i = 0$, i -й человек уходит со встречи сразу после ее начала.

Встреча считается тем более *продуктивной*, чем больше в течение нее состоялось разговоров.

По данному массиву общительности a определите, какие люди должны разговаривать друг с другом, чтобы суммарное количество разговоров было как можно больше.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В следующих $2t$ строках даны описания наборов входных данных.

В описании каждого набора данных первая строка содержит целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество людей на встрече, а во второй строке через пробел перечислены n целых чисел a_1, a_2, \dots, a_n ($0 \leq a_i \leq 2 \cdot 10^5$) — параметры общительности всех людей.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$. Также гарантируется, что сумма всех a_i (по всем наборам входных данных и всем i) не превосходит $2 \cdot 10^5$.

Выходные данные

Выведите t ответов на все наборы входных данных.

В первой строке ответа выведите число k — максимальное количество разговоров, которое возможно провести за встречу.

В каждой из следующих k строк выведите через пробел по два целых числа i и j ($1 \leq i, j \leq n$ и $i \neq j$) — номера людей, между которыми состоится очередной разговор.

Если ответов несколько, выведите любой из них.

Пример

входные данные	Скопировать
8	
2	
2 3	
3	
1 2 3	
4	
1 2 3 4	
3	
0 0 2	
2	
6 2	
3	
0 0 2	
5	
8 2 0 1 1	
5	
0 1 0 0 6	
выходные данные	Скопировать
2	
1 2	
1 2	
3	
1 3	
2 3	
2 3	
5	
1 3	

```
2 4
2 4
3 4
3 4
0
2
1 2
1 2
0
4
1 2
1 5
1 4
1 2
1
5 2
```

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:37:21^{UTC+5} (I1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке




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

C. Кефа и парк

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Кефа решил отпраздновать свой первый крупный заработка походом в ресторан.

Он живет возле необычного парка. Парк представляет из себя подвешенное дерево из n вершин с корнем в вершине 1. В вершине 1 также находится дом Кефы. К сожалению для нашего героя, в парке также находятся коты. Кефа уже выяснил номера вершин, в которых находятся коты.

В листовых вершинах парка находятся рестораны. Кефа хочет выбрать ресторан, в который он пойдет, но, к сожалению, он очень боится котов, поэтому он ни за что не пойдет в ресторан, на пути к которому от его дома найдется более m **подряд идущих** вершин с котами.

Ваша задача — помочь Кефе посчитать количество ресторанов, в которые он может сходить.

Входные данные

В первой строке записаны два целых числа n и m ($2 \leq n \leq 10^5$, $1 \leq m \leq n$) — количество вершин дерева и максимальное количество подряд идущих вершин с котами, которое способен перенести Кефа.

Во второй строке содержится n целых чисел a_1, a_2, \dots, a_n , где каждое a_i либо равняется 0 (тогда в вершине i нет кота), либо равняется 1 (тогда в вершине i есть кот).

В следующих $n - 1$ строках записаны ребра дерева в формате « $x_i y_i$ »(без кавычек) ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$), где x_i и y_i — вершины дерева, соединенные очередным ребром.

Гарантируется, что данный набор рёбер задаёт дерево.

Выходные данные

Одно целое число — количество различных листьев дерева, на пути от дома Кефы до которых не больше m подряд идущих вершин с котами.

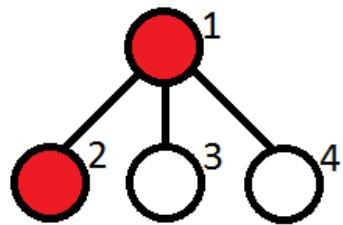
Примеры

входные данные	Скопировать
4 1 1 1 0 0 1 2 1 3 1 4	
выходные данные	Скопировать
2	
входные данные	Скопировать
7 1 1 0 1 1 0 0 0 1 2 1 3 2 4 2 5 3 6 3 7	
выходные данные	Скопировать
2	

Примечание

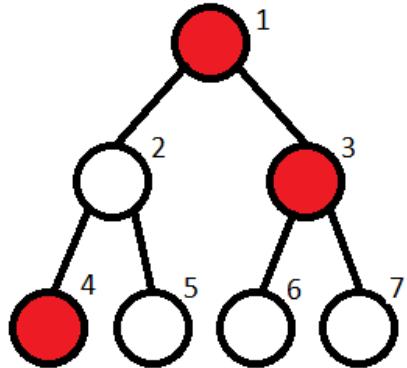
Напомним, что дерево — это связный граф на n вершинах, состоящий из $n - 1$ ребра. Подвешенное дерево — это дерево с особой выделенной вершиной, корнем. В подвешенном дереве из любых двух вершин, соединённых ребром, одна называется предком (та из них, что ближе к корню), а оставшаяся — ребёнком. Вершина называется листом, если у неё нет детей.

Пояснение к первому тесту из условия:



Красным цветом отмечены вершины в которых находятся коты. Рестораны находятся в вершинах 2, 3, 4. Кефа не может сходить только в ресторан, находящийся в вершине 2.

Пояснение ко второму тесту из условия:



Рестораны находятся в вершинах 4, 5, 6, 7. Кефа не может попасть в рестораны 6, 7.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:37:27^{UTC+5} (I1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке




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

E. Компоненты-циклы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Вам задан неориентированный граф, состоящий из n вершин и m ребер. Ваша задача — найти количество компонент связности, которые являются циклами.

Вот несколько определений из теории графов.

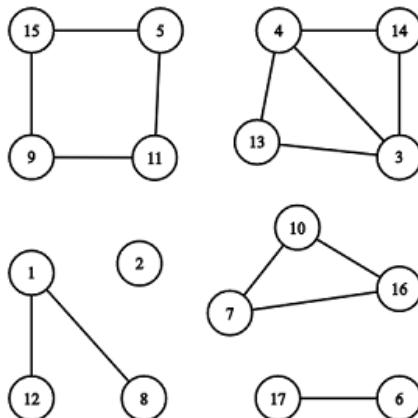
Неориентированный граф состоит из двух множеств: множества узлов (называемых вершинами) и множества рёбер. Каждое ребро соединяет пару вершин. Все ребра двунаправленные (то есть если вершина a соединена с вершиной b , вершина b тоже соединена с вершиной a). Ребро не может соединять вершину саму с собой, также не может существовать более одного ребра между парой вершин.

Две вершины u и v принадлежат одной компоненте связности тогда и только тогда, когда существует хотя бы один путь по ребрам, соединяющий u и v .

Компонента связности является циклом тогда и только тогда, когда все ее вершины могут быть переупорядочены следующим образом:

- первая вершина соединена ребром со второй вершиной,
- вторая вершина соединена ребром с третьей вершиной,
- ...
- последняя вершина соединена ребром с первой вершиной,
- все описанные выше ребра цикла — различны.

Цикл содержит только вершины и ребра, описанные выше. По определению цикл содержит не менее трех вершин.



Граф на рисунке содержит 6 компонент связности, 2 из них являются циклами: [7, 10, 16] и [5, 11, 9, 15].

Входные данные

В первой строке входных данных задано два целых числа n и m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 2 \cdot 10^5$) — количество вершин и рёбер в графе.

В следующих m строках заданы рёбра: i -е ребро задаётся парой вершин v_i, u_i ($1 \leq v_i, u_i \leq n$, $u_i \neq v_i$). Гарантируется, что граф не содержит кратных рёбер (то есть для любой пары (v_i, u_i) не существует других пар (v_i, u_i) и (u_i, v_i) среди заданных рёбер).

Выходные данные

Выведите одно целое число — количество компонент связности, которые являются циклами.

Примеры

входные данные	Скопировать
<pre>5 4 1 2 3 4 5 4 3 5</pre>	

выходные данные

1

входные данные

```
17 15
1 8
1 12
5 11
11 9
9 15
15 5
4 13
3 13
4 3
10 16
7 10
16 7
14 3
14 4
17 6
```

выходные данные

2

Примечание

В первом тестовом примере только компонента $[3, 4, 5]$ является циклом.

Второй пример соответствует рисунку из условия.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:37:30 UTC+5 (11).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО


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

B. Забег за золотом

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Олимпийские игры только что начались, и Федерико с нетерпением ждет начала марафонского забега.

В марафоне будут участвовать n спортсменов, от 1 до n , и все они в прошлом участвовали в 5 важных марафонах, пронумерованных от 1 до 5. Для каждого $1 \leq i \leq n$ и $1 \leq j \leq 5$ Федерико помнит, что спортсмен i занял место $r_{i,j}$ в марафоне j (например, $r_{2,4} = 3$ означает, что спортсмен 2 был третьим в марафоне 4).

Федерико считает, что спортсмен x превосходит спортсмена y , если спортсмен x занял лучшее место, чем спортсмен y , по крайней мере в 3 прошлых марафонах, т.е. $r_{x,j} < r_{y,j}$ по крайней мере для 3 различных значений j .

Федерико считает, что у спортсмена хорошие шансы получить золотую медаль на Олимпийских играх, если он превосходит всех остальных спортсменов.

Найдите любого спортсмена, у которого хорошие шансы получить золотую медаль (то есть спортсмена, который превосходит всех остальных спортсменов), или определите, что такого спортсмена нет.

Входные данные

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

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 50\,000$) — количество спортсменов.

Затем следуют n строк, каждая из которых описывает места одного спортсмена.

i -я из этих строк содержит 5 целых чисел $r_{i,1}, r_{i,2}, r_{i,3}, r_{i,4}, r_{i,5}$ ($1 \leq r_{i,j} \leq 50\,000$) — места спортсмена i в последних 5 марафонах. Гарантируется, что в каждом из 5 прошедших марафонов n спортсменов заняли различные места, т.е. для каждого $1 \leq j \leq 5$, n значений $r_{1,j}, r_{2,j}, \dots, r_{n,j}$ попарно различны.

Гарантируется, что сумма n по всем наборам входных данных не превышает 50 000.

Выходные данные

Для каждого набора входных данных выведите одно целое число — номер спортсмена, у которого хорошие шансы получить золотую медаль (то есть спортсмена, который превосходит всех остальных спортсменов). Если таких спортсменов нет, выведите -1 . Если таких спортсменов больше, чем один, выведите любого из них.

Пример

входные данные	Скопировать
4	
1	
50000 1 50000 50000 50000	
3	
10 10 20 30 30	
20 20 30 10 10	
30 30 10 20 20	
3	
1 1 1 1 1	
2 2 2 2 2	
3 3 3 3 3	
6	
9 5 3 7 1	
7 4 1 6 8	
5 6 7 3 2	
6 7 8 8 6	
4 2 2 4 5	
8 3 6 9 4	
выходные данные	Скопировать
1	
-1	
1	
5	

Примечание

Объяснение первого набора входных данных: Есть только один спортсмен, поэтому он превосходит всех остальных (так как больше никого нет), и поэтому у него хорошие шансы получить золотую медаль.

Пояснение второго набора входных данных: Есть $n = 3$ спортсменов.

- Спортсмен 1 превосходит спортсмена 2. Действительно, спортсмен 1 лучше спортсмена 2 в марафонах 1, 2 и 3.
- Спортсмен 2 превосходит спортсмена 3. Действительно, спортсмен 2 превосходит спортсмена 3 в марафонах 1, 2, 4 и 5.
- Спортсмен 3 превосходит спортсмена 1. Действительно, спортсмен 3 лучше, чем спортсмен 1 в марафонах 3, 4 и 5.

Объяснение третьего набора входных данных: Есть $n = 3$ спортсменов.

- Спортсмен 1 превосходит спортсменов 2 и 3. Поскольку он превосходит всех остальных спортсменов, у него хорошие шансы получить золотую медаль.
- Спортсмен 2 превосходит спортсмена 3.
- Спортсмен 3 не превосходит ни одного другого спортсмена.

Пояснение четвертого набора входных данных: Есть $n = 6$ спортсменов.

- Спортсмен 1 превосходит спортсменов 3, 4, 6.
- Спортсмен 2 превосходит спортсменов 1, 4, 6.
- Спортсмен 3 превосходит спортсменов 2, 4, 6.
- Спортсмен 4 не превосходит ни одного другого спортсмена.
- Спортсмен 5 превосходит спортсменов 1, 2, 3, 4, 6. Поскольку он превосходит всех остальных спортсменов, у него хорошие шансы получить золотую медаль.
- Спортсмен 6 превосходит только спортсмена 4.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:37:32_{UTC+5} (I1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке



ИТМО


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

A. Мишка и условие дружбы

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Мишка Лимак исследует социальную сеть. Основная ее функциональность заключается в том, что два пользователя могут стать друзьями (в таком случае они могут обмениваться сообщениями и смешными картинками).

Сейчас в социальной сети n пользователей, они пронумерованы от 1 до n . Ровно m пар из них являются друзьями. Конечно, никакой пользователь не может быть другом с самим собой.

Пусть запись A-B означает, что пользователи A and B друзья. Лимак считает, что социальная сеть обоснована, если и только если выполняется следующее условие: для любых трех **различных** пользователей (X, Y, Z) если X-Y и Y-Z, то также X-Z.

Например, если Алан и Боб друзья, и Боб и Цири друзья, то Алан и Цири тоже должны быть друзьями.

Можете помочь Лимаку и проверить, что социальная сеть обоснована? Выведите «YES» или «NO» без кавычек в соответствии с результатом.

Входные данные

Первая строка содержит два целых числа n и m ($3 \leq n \leq 150\,000$, $0 \leq m \leq \min(150\,000, \frac{n \cdot (n-1)}{2})$) — число пользователей сети и число пар пользователей, которые являются друзьями.

i -я из следующих m строк содержит два целых различных числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), означающих, что a_i -й и b_i -й пользователи являются друзьями. Все пары друзей различны.

Выходные данные

Если данная социальная сеть обоснована, выведите в единственную строку «YES» (без кавычек).

Иначе выведите в единственную строку «NO» (без кавычек).

Примеры

входные данные <pre>4 3 1 3 3 4 1 4</pre>	<input type="button" value="Скопировать"/>
выходные данные <pre>YES</pre>	<input type="button" value="Скопировать"/>
входные данные <pre>4 4 3 1 2 3 3 4 1 2</pre>	<input type="button" value="Скопировать"/>
выходные данные <pre>NO</pre>	<input type="button" value="Скопировать"/>
входные данные <pre>10 4 4 3 5 10 8 9 1 2</pre>	<input type="button" value="Скопировать"/>
выходные данные <pre>YES</pre>	<input type="button" value="Скопировать"/>
входные данные <pre>3 2 1 2 2 3</pre>	<input type="button" value="Скопировать"/>

[Скопировать](#)**выходные данные**

NO

Примечание

Рисунки ниже показывают социальную сеть в первом (слева) и во втором (справа) примерах. Кружки обозначают пользователей, а линии означают, что два пользователя являются друзьями.

Во втором примере ответ «NO», потому что пользователи (2, 3) друзья, пользователи (3, 4) тоже друзья, а пользователи (2, 4) — нет.



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:37:38^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ITMO


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

C. Циклические перестановки

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Перестановка длины n — это массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — это перестановка, но $[1, 2, 2]$ — это не перестановка (2 встречается дважды в массиве), а $[1, 3, 4]$ также не является перестановкой ($n = 3$, но в массиве встречается 4).

Рассмотрим перестановку p длины n . Построим граф на n вершинах, используя перестановку следующим образом:

- Для каждого $1 \leq i \leq n$ найдите **наибольшее** значение j , для которого $1 \leq j < i$ и $p_j > p_i$, и добавьте неориентированное ребро между вершинами i и j .
- Для каждого $1 \leq i \leq n$ найдите **наименьшее** значение j , для которого $i < j \leq n$ и $p_j > p_i$, и добавьте неориентированное ребро между вершинами i и j

В тех случаях, когда таких j не существует, мы не добавляем ребер. Также обратите внимание, что мы проводим ребра между соответствующими индексами, а не значениями в этих индексах.

Например, рассмотрим случай $n = 4$ и $p = [3, 1, 4, 2]$; здесь ребрами графа являются $(1, 3), (2, 1), (2, 3), (4, 3)$.

Перестановка p является **циклической**, если граф, построенный с использованием p , имеет хотя бы один простой цикл.

Для данного n , найдите число циклических перестановок длины n . Поскольку число может быть очень большим, выведите его по модулю $10^9 + 7$.

Пожалуйста, обратитесь к разделу Примечания для формального определения простого цикла.

Входные данные

Первая и единственная строка содержит одно целое число n ($3 \leq n \leq 10^6$).

Выходные данные

Выведите единственное целое число $0 \leq x < 10^9 + 7$, количество циклических перестановок длины n по модулю $10^9 + 7$.

Примеры

входные данные	<input type="button" value="Скопировать"/>
4	
выходные данные	<input type="button" value="Скопировать"/>
16	
входные данные	<input type="button" value="Скопировать"/>
583291	
выходные данные	<input type="button" value="Скопировать"/>
135712853	

Примечание

Для $n = 4$ существует 16 циклических перестановок. $[4, 2, 1, 3]$ — одна из таких перестановок, она содержит цикл длины четыре: $4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 4$.

Вершины v_1, v_2, \dots, v_k образуют простой цикл, если выполняются следующие условия:

- $k \geq 3$.
- $v_i \neq v_j$ для любой пары индексов i и j . ($1 \leq i < j \leq k$)
- Между v_i и v_{i+1} есть ребро для всех i ($1 \leq i < k$), как и между v_1 и v_k


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

D. Лунный новый год и прогулка

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

Приближается лунный новый год, а Боб решил отправиться на прогулку в ближайший парк.

Парк может быть представлен как связный граф из n вершин и m неориентированных ребер. Изначально Боб находился в вершине 1 и записал 1 в свою записную книжку. Он может переходить из одной вершины в другую по данным ребрам. Каждый раз, когда он посещает вершину, еще не записанную в его книжку, он записывает ее. После того, как он посетит все вершины как минимум по разу, он закончит прогулку, а в его книжке будет записана перестановка вершин a_1, a_2, \dots, a_n .

Гулять скучно, а решать задачи — интересно. Боб хочет узнать лексикографически минимальную перестановку, которую он может получить по итогам прогулки. Бобу эта задача кажется простой, поэтому он отдал ее вам.

Последовательность x лексикографически меньше последовательности y , если и только если выполняется один из следующих пунктов:

- x — префикс y , но $x \neq y$ (обратите внимание, в этой задаче такое невозможно, так как все рассматриваемые последовательности имеют одинаковую длину);
- в первой позиции, где x и y различны, в последовательности x элемент меньше, чем соответствующий элемент в y .

Входные данные

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 10^5$) — число вершин и ребер в графе, соответственно.

Следующие m строк описывают неориентированные ребра графа. i -я из этих строк содержит два целых числа u_i и v_i ($1 \leq u_i, v_i \leq n$) — вершины, соединенные i -м ребром.

Обратите внимание, что в графе могут быть кратные ребра и петли. Гарантируется, что граф связный.

Выходные данные

Выведите лексикографически наименьшую последовательность a_1, a_2, \dots, a_n из тех, которые может записать Боб.

Примеры

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

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

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

выходные данные

1 4 3 7 9 8 6 5 2 10

[Скопировать](#)**Примечание**

В первом примере одним из возможных путей Боба является путь $1 \rightarrow 2 \rightarrow 1 \rightarrow 3$. Боб в этом случае запишет последовательность $\{1, 2, 3\}$, которая является лексикографически наименьшей.

Во втором примере Боб может пойти по пути $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 5$. Тогда он запишет последовательность $\{1, 4, 3, 2, 5\}$, которая является лексикографически наименьшей.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:37:40^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке




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

A. NP-трудная задача

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Недавно Пари и Арий узнали про NP-трудные задачи, особенно им понравилась задача о [минимальном вершинном покрытии](#).

Пусть нам дан некоторый граф G . Подмножество A его вершин называется [вершинным покрытием](#), если для любого ребра uv хотя бы один его конец лежит в множестве, то есть выполнено $u \in A$ или $v \in A$ (или оба условия).

Пари и Арий выиграли на командной олимпиаде чудесный неориентированный граф, и теперь каждый хочет забрать себе множество его вершин, являющееся вершинным покрытием.

Они отдали свой граф вам и попросили выбрать два [непересекающихся](#) множества вершин A и B , таких что и A и B являются вершинным покрытием, или определить, что сделать это невозможно. Каждая вершина может быть отдана только одному из друзей (а некоторые и вовсе можно не отдавать никому).

Входные данные

В первой строке входных данных записаны два числа n и m ($2 \leq n \leq 100\,000$, $1 \leq m \leq 100\,000$) — количество вершин и количество рёбер в выигранном друзьями графе.

В каждой из следующих m строк записана пара чисел u_i и v_i ($1 \leq u_i, v_i \leq n$), означающая ненаправленное ребро между вершинами u_i и v_i . Гарантируется, что в графе отсутствуют петли и кратные рёбра.

Выходные данные

Если невозможно поделить граф между Пари и Арием, как они этого хотят, то выведите «`-1`» (без кавычек).

Если же существуют два непересекающихся вершинных покрытия, то выведите их описания. Каждое описание состоит из двух строк. Первая строка должна содержать единственное число k — количество вершин в данном вершинном покрытии, а вторая строка должна содержать k чисел — индексы вершин в покрытии. Обратите внимание, что поскольку $m \geq 1$, никакое вершинное покрытие не может быть пустым.

Примеры

входные данные	<button>Скопировать</button>
<code>4 2</code>	
<code>1 2</code>	
<code>2 3</code>	
выходные данные	<button>Скопировать</button>
<code>1</code>	
<code>2</code>	
<code>2</code>	
<code>1 3</code>	
входные данные	<button>Скопировать</button>
<code>3 3</code>	
<code>1 2</code>	
<code>2 3</code>	
<code>1 3</code>	
выходные данные	<button>Скопировать</button>
<code>-1</code>	

Примечание

В первом примере можно отдать Арию вершину номер 2, а Пари вершины с номерами 1 и 3. Вершину 4 можно оставить себе (а можно тоже кому-нибудь отдать).

Во втором примере не существует способа раздать вершины так, чтобы удовлетворить и Пари, и Ария.

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

C. Как можно меньше ничьих

ограничение по времени на тест: 1 секунда

ограничение по памяти на тест: 256 мегабайт

Скоро состоится большой футбольный чемпионат! n команд будут участвовать в нем, и каждая пара различных команд сыграет ровно один матч друг с другом.

У матча может быть два различных исхода:

- матч может закончиться вничью, тогда обе команды получат по 1 очку;
- одна из команд может победить, тогда победившая команда получит 3 очка, а проигравшая — 0.

Счет команды — суммарное количество очков, полученное командой за все матчи, в которых она играла.

Вас интересует гипотетическая ситуация, в которой все команды будут в **конце чемпионата иметь одинаковый счет**. Простой пример такой ситуации — если все матчи закончатся вничью, но вас интересует подобная ситуация с минимально возможным количеством ничьих.

Ваша задача — найти ситуацию (выбрать результат каждой игры), в которой у всех команд одинаковый итоговый счет, а количество ничьих минимально возможно.

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных.

Затем следуют сами наборы входных данных. Каждый набор описывается одной строкой, содержащей целое число n ($2 \leq n \leq 100$) — количество команд.

Выходные данные

Для каждого набора входных данных выведите $\frac{n(n-1)}{2}$ целых чисел, описывающих результаты игр в следующем порядке: первое число соответствует результату матча между командой 1 и командой 2, второе — между командой 1 и командой 3, затем 1 и 4, ..., 1 и n , 2 и 3, 2 и 4, ..., 2 и n , и так далее, до результата матча между командой $n - 1$ и командой n .

Число, соответствующее результату матча между командой x и командой y , должно быть равно 1, если побеждает команда x , −1, если побеждена команда y , или 0, если матч заканчивается вничью.

Все команды должны иметь одинаковый итоговый счет, и количество ничьих должно быть минимально возможным. Если оптимальных ответов несколько — выведите любой из них. Можно показать, что всегда есть способ составить результаты игр так, что все команды имеют одинаковый итоговый счет.

Пример

входные данные	Скопировать
2 2 3	
выходные данные	Скопировать
0 1 -1 1	

Примечание

В первом наборе входных данных из примера обе команды получают по 1 очку, так как их матч заканчивается вничью.

Во втором наборе входных данных из примера команда 1 побеждает команду 2 (команда 1 получает 3 очка), команда 1 проигрывает команде 3 (команда 3 получает 3 очка), и команда 2 выигрывает у команды 3 (команда 2 получает 3 очка).


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

C. Лиса и имена

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Лиса Ciel собирается опубликовать статью по ЛИС (Лисооперируемым Интеллектуальным Системам). Она слышала, что список авторов научной статьи всегда сортируется в лексикографическом порядке.

После изучения примеров оформления, лиса обнаружила, что иногда это утверждение не верно. В некоторых статьях имена авторов не сортируются в лексикографическом порядке в обычном понимании. Но, оказывается, верно то, что после некоторого изменения порядка букв в алфавите, порядок авторов становится лексикографическим.

Лиса хочет знать, существует ли такой порядок букв латинского алфавита, что имена авторов предлагаемой ею статьи следуют в лексикографическом порядке. Если да, то требуется также найти любой такой порядок.

Лексикографический порядок определяется следующим образом. Мы сравниваем s и t , сперва находя крайнюю левую позицию с различающимися символами: $s_i \neq t_i$. Если такой позиции нет (то есть, s — это префикс t или наоборот), то более короткая строка меньше. В противном случае мы сравниваем символы s_i и t_i согласно их порядку в алфавите.

Входные данные

В первой строке записано целое число n ($1 \leq n \leq 100$), количество имен.

В каждой из следующих n строк записано по одному слову $name_i$ ($1 \leq |name_i| \leq 100$), обозначающему i -е имя. Каждое имя содержит только строчные буквы латинского алфавита. Все имена различны.

Выходные данные

Если существует такой порядок букв, при котором имена в данном списке следуют в лексикографическом порядке, выведите любой такой порядок в виде перестановки символов 'a'–'z' (иными словами, выведите сначала первую букву модифицированного алфавита, затем вторую, и так далее).

В противном случае выведите единственное слово «Impossible» (без кавычек).

Примеры

входные данные	Скопировать
3 rivest shamir adleman	
выходные данные	Скопировать
bcd e fghijklmnpqrstuvwxyz	

входные данные	Скопировать
10 tourist petr wjmzbmr yeputons vepifanov scottwu oooooooooooooooooo subscriber rowdark tankengineer	
выходные данные	Скопировать
Impossible	

входные данные	Скопировать
10 petr egor endagorion feferivan ilovetanyaromanova kostka dmitriyh	

maratsnowbear
bredorjaguarturnik
cgyforever

ВЫХОДНЫЕ ДАННЫЕ**Скопировать**

aghjlnopefikdmЬсqrstuvwxyz

ВХОДНЫЕ ДАННЫЕ**Скопировать**

7
car
care
careful
carefully
becarefuldontforgetsomething
otherwiseyouwillbehacked
goodluck

ВЫХОДНЫЕ ДАННЫЕ**Скопировать**

acbdefhijklmnogpqrstuvwxyz

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:38:32^{UTC+5} (I1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#).

При поддержке



ИТМО


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

E. Раздели на два набора

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Поликарпу недавно подарили набор из n (число n — чётное) костяшек домино. На каждой костяшке написаны два целых числа от 1 до n .

Сможет ли он разделить все костяшки на два набора так, чтобы все числа на костяшках каждого набора были различны? Каждая костяшка должна войти ровно в один из двух наборов.

Например, если у него есть 4 костяшки: $\{1, 4\}$, $\{1, 3\}$, $\{3, 2\}$ и $\{4, 2\}$, то Поликарп сможет разделить их на два набора требуемым образом. В первый набор может войти первая и третья костяшка ($\{1, 4\}$ и $\{3, 2\}$), а во второй — вторая и четвёртая ($\{1, 3\}$ и $\{4, 2\}$).

Входные данные

В первой строке записано единственное число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Далее следуют описания наборов входных данных.

В первой строке каждого набора входных данных содержится единственное целое чётное число n ($2 \leq n \leq 2 \cdot 10^5$) — количество костяшек.

В следующих n строках содержатся пары чисел a_i и b_i ($1 \leq a_i, b_i \leq n$), описывающие числа на i -й костяшке.

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

Выходные данные

Для каждого набора входных данных выведите:

- YES, если возможно разделить n костяшек на два набора так, что числа на костяшках каждого набора различны;
- NO, если сделать это невозможно.

Вы можете выводить YES и NO в любом регистре (например, строки yEs, yes, Yes и YES будут распознаны как положительный ответ).

Пример

входные данные	Скопировать
<pre>6 4 1 2 4 3 2 1 3 4 6 1 2 4 5 1 3 4 6 2 3 5 6 2 1 1 2 2 2 1 2 2 1 8 2 1 1 2 4 3 4 3 5 6 5 7 8 6 7 8 8 1 2 2 1 4 3</pre>	

```
5 3
5 4
6 7
8 6
7 8
```

ВХОДНЫЕ ДАННЫЕ**Скопировать**

```
YES
NO
NO
YES
YES
NO
```

Примечание

В первом наборе входных данных костяшки можно разделить так:

- Первый набор: $[\{1, 2\}, \{4, 3\}]$
- Второй набор: $[\{2, 1\}, \{3, 4\}]$

Иными словами в первый набор берём костяшки с номерами 1 и 2, а во второй набор берём костяшки с номерами 3 и 4.

Во втором наборе входных данных как костяшки не раздели на 2 набора, хотя бы в одном будут повторяющиеся числа.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:38:34_{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО

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

D. Дерево минимальной высоты

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

У Монокарпа было дерево из n вершин с корнем в вершине 1. Он решил изучить алгоритм BFS (Поиск в ширину), и поэтому запустил BFS на своем дереве, стартуя с корня. Упрощенно, BFS можно описать следующим псевдокодом:

```
a = [] # порядок, в котором вершины обработались
q = Queue()
q.put(1) # кладем корень в конец очереди
while not q.empty():
    k = q.pop() # достаем вершину из начала очереди
    a.append(k) # кладем k в конец последовательности, описывающей порядок посещения вершин
    for y in g[k]: # g[k] – это список всех детей вершины k, отсортированный в порядке возрастания
        q.put(y)
```

Монокарп был так восхищен алгоритмом BFS, что в результате потерял свое дерево. К счастью, у него осталась последовательность вершин в порядке их посещения алгоритмом BFS (массив a из псевдокода). Монокарп знает, что каждая вершина была посещена ровно один раз (так как они кладутся и забираются из очереди ровно один раз). Также он знает, что все дети каждой вершины были просмотрены в порядке возрастания.

Монокарп понимает, что есть много деревьев (в общем случае) с одинаковым порядком посещения a , поэтому даже не надеется восстановить свое дерево. Монокарпа устроит любое дерево **минимально возможной высоты**.

Высота дерева — это максимум среди глубин вершин дерева, и глубина вершины — это количество дуг на пути от корня до этой вершины. Например, глубина вершины 1 равна 0, так как она является корнем, а глубина любого из детей корня равна 1.

Помогите Монокарпу найти любое дерево с заданным порядком посещения a и минимальной высотой.

Входные данные

В первой строке задано единственное целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В первой строке каждого набора задано одно целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве.

Во второй строке каждого набора заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$; $a_i \neq a_j$; $a_1 = 1$) — порядок посещения вершин алгоритмом BFS.

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

Выходные данные

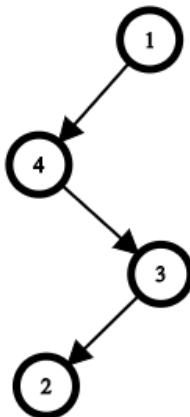
Для каждого набора входных данных, выведите минимально возможную высоту дерева с заданным порядком обхода a .

Пример

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

Примечание

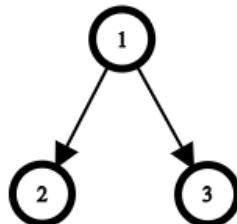
В первом наборе входных данных, есть только одно дерево с заданным порядком посещения:



Во втором наборе, также есть только одно дерево с заданным порядком посещения:



В третьем наборе, оптимальное дерево с заданным порядком посещения изображено ниже:



[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:38:37^{UTC+5} (1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке




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

E. Хороводы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

n людей пришли на праздник и решили станцевать несколько хороводов. В хороводе не менее 2-х людей и у каждого человека есть ровно два соседа, если в хороводе 2 человека, с обоих сторон один и тот же сосед.

Вы решили узнать, сколько именно было хороводов. Но каждый участник праздника запомнил ровно **одного** соседа. Ваша задача — определить, какое минимальное и максимальное число хороводов могло быть.

Например, если на празднике было 6 человек, и номера соседей, которых они запомнили, равны [2, 1, 4, 3, 6, 5] соответственно, то хороводов могло быть **минимум 1**:

- 1 – 2 – 3 – 4 – 5 – 6 – 1

и **максимум 3**:

- 1 – 2 – 1
- 3 – 4 – 3
- 5 – 6 – 5

Входные данные

Первая строка содержит положительное число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка описания каждого набора входных данных содержит положительное число n ($2 \leq n \leq 2 \cdot 10^5$) — количество людей на празднике.

Вторая строка описания каждого набора входных данных содержит n целых чисел a_i ($1 \leq a_i \leq n, a_i \neq i$) — номер соседа, которого запомнил i -й человек.

Гарантируется, что входные данные корректны и соответствуют хотя бы одному разбиению людей на хороводы.

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

Выходные данные

Для каждого набора входных данных выведите два целых числа — минимальное и максимальное количество хороводов, которое могло быть.

Пример

входные данные

[Скопировать](#)

```
10
6
2 1 4 3 6 5
6
2 3 1 5 6 4
9
2 3 2 5 6 5 8 9 8
2
2 1
4
4 3 2 1
5
2 3 4 5 1
6
5 3 4 1 1 2
5
3 5 4 1 2
6
6 3 2 5 4 3
6
5 1 4 3 4 2
```

выходные данные

[Скопировать](#)

```
1 3
2 2
1 3
```

```
1 1
1 2
1 1
1 1
2 2
1 2
1 1
```

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов
Соревнования по программированию 2.0
Время на сервере: 29.12.2024 15:38:50^{UTC+5} (1).
Мобильная версия, переключиться на [десктопную](#).
[Privacy Policy](#)

При поддержке




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

A. Два пути

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

В Абсурдистане n городов (пронумерованных от 1 до n) и m двунаправленных железнодорожных перегонов. А ещё там абсурдно простая сеть автотрасс — для любой пары различных городов x и y между ними есть двунаправленная автомобильная дорога **тогда и только тогда**, когда между ними нет железнодорожного перегона. На перемещение в соседний город по одному железнодорожному перегону или по одной автомобильной дороге уходит ровно один час.

Поезд и автобус одновременно выезжают из города 1. У них обоих один и тот же пункт назначения, город n , и они нигде не останавливаются по пути (кроме итоговой стоянки в городе n). Поезд перемещается только по железнодорожным перегонам, а автобус только по автомобильным дорогам.

Вас попросили составить маршруты для обоих транспортных средств; каждый маршрут может проходить по любой дороге/перегону произвольное количество раз. Одним из наиболее важных пунктов при составлении маршрута является безопасность, поэтому, во избежании несчастных случаев на железнодорожных переездах, поезд и автобус никогда не должны одновременно прибывать в один и тот же город, кроме, возможно, города n .

Через какое наименьшее количество часов оба транспортных могут быть в городе n (максимальное из времён прибытия для автобуса и поезда)? Обратите внимание, что автобус и поезд не обязаны прибыть в город n одновременно, но могут так сделать, если потребуется.

Входные данные

Первая строка входных данных содержит два числа n и m ($2 \leq n \leq 400$, $0 \leq m \leq n(n - 1) / 2$) — количество городов и количество железнодорожных перегонов соответственно.

В каждой из следующих m строк записано по два целых числа u и v , обозначающих наличие железнодорожного перегона между городами u и v ($1 \leq u, v \leq n$, $u \neq v$).

Гарантируется, что любые два города соединены не более чем одним железнодорожным перегоном.

Выходные данные

Выведите единственное целое число — наименьшее возможное время прибытия последнего из транспортных средств в город n . Если хотя бы одно из двух транспортных средств не сможет добраться до города n , то выведите - 1.

Примеры

входные данные	Скопировать
4 2 1 3 3 4	

выходные данные	Скопировать
2	

входные данные	Скопировать
4 6 1 2 1 3 1 4 2 3 2 4 3 4	

выходные данные	Скопировать
-1	

входные данные	Скопировать
5 5 4 2 3 5 4 5 5 1 1 2	

выходные данные**Скопировать**

3

Примечание

В первом примере поезд может поехать по маршруту $1 \rightarrow 3 \rightarrow 4$, а автобус по маршруту $1 \rightarrow 2 \rightarrow 4$. Обратите внимание, что автобусу и поезду разрешается прибывать в город 4 одновременно.

Во втором примере Абсурдистаном правят железнодорожники. Обычных дорог (не ЖД перегонов) нет, так что автобус никак не может добраться до города 4.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:38:53^{UTC+5} (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО

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

D. Стока минимальной стоимости

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

Определим стоимость строки s как количество пар индексов i и j ($1 \leq i < j < |s|$) таких, что $s_i = s_j$ и $s_{i+1} = s_{j+1}$.

Заданы два положительных целых числа n и k . Среди всех строк длины n , содержащих только первые k букв латинского алфавита, найдите строку с минимально возможной стоимостью. Если таких строк несколько — найдите любую из них.

Входные данные

Единственная строка содержит два целых числа n и k ($1 \leq n \leq 2 \cdot 10^5$; $1 \leq k \leq 26$).

Выходные данные

Выведите строку s такую, что она состоит из n символов, каждый из которых является одной из первых k латинских букв, и она имеет минимально возможную стоимость среди всех таких строк. Если таких строк несколько — выведите любую из них.

Примеры

входные данные	Скопировать
9 4	
выходные данные	Скопировать
aabacadb	
входные данные	Скопировать
5 1	
выходные данные	Скопировать
aaaaa	
входные данные	Скопировать
10 26	
выходные данные	Скопировать
codeforces	

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:38:55 UTC+5 (I1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО

C. Наидлиннейший простой цикл

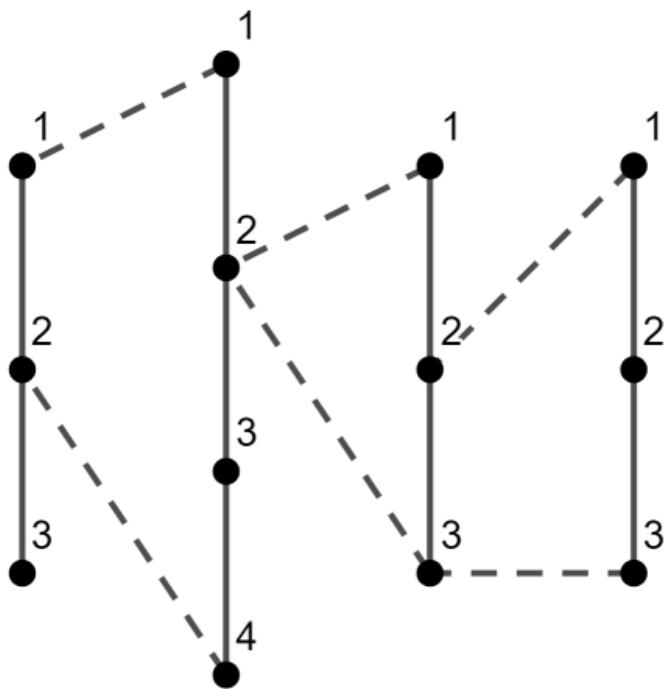
ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

У вас есть n цепей, и i -я цепь состоит из c_i вершин. Вершины в каждой цепи пронумерованы независимо от 1 по c_i вдоль этой цепи. Другими словами, i -я цепь — это неориентированный граф из c_i вершин и $(c_i - 1)$ ребер, соединяющих j -ю и $(j + 1)$ -ю вершины для всех $1 \leq j < c_i$.

Вы решили объединить эти цепи в один граф следующим образом:

1. первая цепь пропускается;
2. 1-я вершина i -й цепи соединяется ребром с a_i -й вершиной $(i - 1)$ -й цепи;
3. последняя (c_i -я) вершина i -й цепи соединяется ребром с b_i -й вершиной $(i - 1)$ -й цепи.



Изображение первого набора входных данных. Пунктирные линии означают ребра, добавленные во время объединения
Посчитайте длину найдлиннейшего простого цикла в полученном графе.

Простой цикл — это, фактически, цепь, в которой первая и последняя вершины также соединены ребром. Если двигаться вдоль простого цикла, то мы посетим каждую вершину этого цикла ровно по одному разу.

Входные данные

В первой строке задано одно целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

В первой строке каждого набора задано одно целое число n ($2 \leq n \leq 10^5$) — количество цепей.

Во второй строке каждого набора заданы n целых чисел c_1, c_2, \dots, c_n ($2 \leq c_i \leq 10^9$) — количество вершин в соответствующих цепях.

В третьей строке каждого набора заданы n целых чисел a_1, a_2, \dots, a_n ($a_1 = -1; 1 \leq a_i \leq c_{i-1}$).

В четвертой строке каждого набора заданы n целых чисел b_1, b_2, \dots, b_n ($b_1 = -1; 1 \leq b_i \leq c_{i-1}$).

Оба числа a_1 и b_1 равны -1 , они не используются в построении графа и заданы лишь для соответствия нумерации.
Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 .

Выходные данные

Для каждого набора входных данных, выведите длину наибольшего простого цикла.

Пример

входные данные**Скопировать**

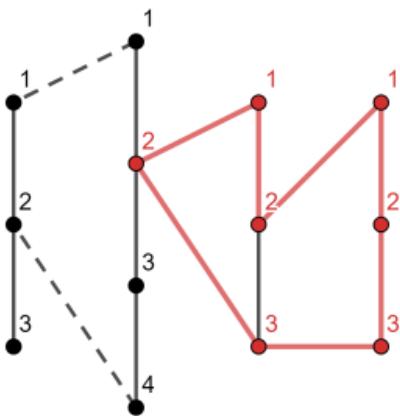
```
3
4
3 4 3 3
-1 1 2 2
-1 2 2 3
2
5 6
-1 5
-1 1
3
3 5 2
-1 1 1
-1 3 5
```

выходные данные**Скопировать**

```
7
11
8
```

Примечание

В первом наборе входных данных, найденнейший простой цикл изображен ниже:



Мы не можем увеличить этот цикл с помощью первой цепи, так как в этом случае цикл уже не будет простым — вершина 2 на второй цепи нарушит простоту цикла.

[Codeforces](#) (c) Copyright 2010-2024 Михаил Мирзаянов

Соревнования по программированию 2.0

Время на сервере: 29.12.2024 15:38:57_{UTC+5} (1).

Мобильная версия, переключиться на [десктопную](#).

[Privacy Policy](#)

При поддержке



ИТМО


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

D. Цветные порталы

ограничение по времени на тест: 2 секунды

ограничение по памяти на тест: 256 мегабайт

На прямой расположено n городов. Города пронумерованы целыми числами от 1 до n .

Для перемещения между городами используются порталы. Порталы бывают 4 цветов: синий, зеленый, красный и желтый. В каждом городе расположены порталы двух разных цветов. Из города i можно переместиться в город j , если в них есть порталы одинакового цвета (например, между городом «красный-синий» и городом «синий-зеленый» можно перемещаться). Данное перемещение стоит $|i - j|$ монет.

Ваша задача — ответить на q независимых запросов: посчитайте минимальную стоимость, чтобы переместиться из города x в город y .

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 q ($1 \leq n, q \leq 2 \cdot 10^5$) — количество городов и количество запросов, соответственно.

Вторая строка содержит n строк одного из следующих типов: BG, BR, BY, GR, GY или RY; i -я из них описывает порталы, находящиеся в i -м городе; символ B обозначает, что в городе есть портал синего цвета, G — зеленого, R — красного, а Y — желтого.

j -я из следующих q строк содержит два целых числа x_j и y_j ($1 \leq x_j, y_j \leq n$) — описание j -го запроса.

Дополнительные ограничения на входные данные:

- сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$;
- сумма q по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Выходные данные

Для каждого запроса выведите одно целое число — минимальную стоимость, чтобы переместиться из города x в город y (или -1 , если это невозможно).

Пример

входные данные	Скопировать
<pre>2 4 5 BR BR GY GR 1 2 3 1 4 4 1 4 4 2 2 1 BG RY 1 2</pre>	
выходные данные	Скопировать
<pre>1 4 0 3 2 -1</pre>	