Задача A. Паросочетание

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Двудольным графом называется неориентированный граф (V, E), $E \subseteq V \times V$ такой, что его множество вершин V можно разбить на два множества A и B, для которых $\forall (e_1, e_2) \in E$ $e_1 \in A$, $e_2 \in B$ и $A \cup B = V$, $A \cap B = \varnothing$.

Паросочетанием в двудольном графе называется любой набор его несмежных рёбер, то есть такой набор $S \subseteq E$, что для любых двух рёбер $e_1 = (u_1, v_1)$, $e_2 = (u_2, v_2)$ из S $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочетание в двудольном графе, то есть паросочетание с максимально возможным числом рёбер.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leqslant n, m \leqslant 250$), где n — число вершин в множестве A, а m — число вершин в B.

Далее следуют n строк с описаниями рёбер — i-я вершина из A описана в $(i+1)$-й строке файла. Каждая из этих строк содержит номера вершин из B, соединённых с i-й вершиной A. Гарантируется, что в графе нет кратных ребер. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

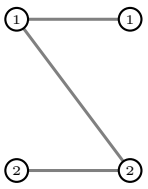
Формат выходных данных

Первая строка выходного файла должна содержать одно целое число l — количество рёбер в максимальном паросочетании. Далее следуют l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы рёбер паросочетания в A и B соотвественно.

Пример

стандартный ввод	стандартный вывод
2 2 1 2 0 2 0	2 1 1 2 2

Замечание



Задача B. Покрытие путями

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Задан ориентированный ациклический граф. Требуется определить минимальное количество не пересекающихся по вершинам путей, покрывающих все вершины.

Формат входных данных

Первая строка входного файла содержит целые числа n и m — количества вершин и рёбер графа соответственно ($2 \leqslant n \leqslant 1\,000$, $0 \leqslant m \leqslant 10^5$). В следующих m строках содержатся по два натуральных числа — номера вершин u и v, которые соединяет ребро (u, v).

Формат выходных данных

В первой строке выходного файла выведите натуральное число k — минимальное количество путей, необходимых для покрытия всех вершин.

Пример

стандартный ввод	стандартный вывод
3 3 1 3 3 2 1 2	1

Задача C. Замощение доминошками

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Дано игровое поле размера $n \times m$, некоторые клетки которого уже замощены. Замостить свободные соседние клетки поля доминошкой размера 1×2 стоит a условных единиц, а замостить свободную клетку поля квадратиком размера 1×1 — b условных единиц.

Определите, какая минимальная сумма денег нужна, чтобы замостить всё поле.

Формат входных данных

Первая строка входного файла содержит 4 целых числа n, m, a, b ($1 \leqslant n, m \leqslant 100$, $|a| \leqslant 1\,000$, $|b| \leqslant 1\,000$). Каждая из последующих n строк содержит по m символов: символ '".“ (точка) обозначает занятую клетку поля, а символ '"*“ (звёздочка) — свободную.

Формат выходных данных

В выходной файл выведите одно число — минимальную сумму денег, имея которую можно замостить свободные клетки поля (и только их).

Пример

стандартный ввод	стандартный вывод
2 3 3 2 .** .*.	5

Задача D. Такси

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Управлять службой такси — сосвем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x-координатой и y-координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d), равно $|a - c| + |b - d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Формат входных данных

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Формат выходных данных

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Примеры

стандартный ввод	стандартный вывод
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

Задача E. Фруктовый сок

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Степан недавно купил себе новую соковыжималку. Теперь по утрам он и его братья и сестры пьют свежевыжатый фруктовый сок. А это, между прочим, очень полезно!

Недавно они поняли, что можно пить сок, выжатый не только из одного вида фруктов, как, например, апельсиновый, но и различные смеси, например, виноградно-яблочный.

В семье Степана все очень любят сок, поэтому могут утром выпить не один стакан, причем разных видов сока. Например, его сестра Катя очень любит грейпфрутовый и апельсиновый соки. Степан, как наиболее технически грамотный человек, каждое утро занимается приготовлением соков.

Опишем подробнее, как работает соковыжималка. В нее загружаются фрукты, они проходят отжим в центрифуге, обезвоженная мякоть сбрасывается в отдельный резервуар, а сок попадает в специальную емкость.

Основная проблема состоит в том, что эту емкость иногда приходится мыть. Например, если после приготовления апельсинового сока, необходимо приготовить яблочный, то емкость надо мыть, иначе получится апельсиново-яблочный сок. Более формально, пусть сок A состоит из компонентов $a_1, \ldots, a_n$, а сок B – из компонентов $b_1, \ldots, b_m$. Сок B можно готовить после сока A, если любой из компонентов a_i является компонентом сока B (т.е. $\exists j : b_j = a_i$). В противном случае емкость для сока надо помыть.

Степан не очень любит мыть посуду, поэтому хочет мыть емкость как можно меньшее число раз. Помогите ему.

Формат входных данных

Первая строка входного файла содержит количество N различных соков, которые требуется приготовить ($1 \leqslant N \leqslant 300$). Каждая из последующих N строк описывает один из соков. Описание сока состоит из числа k его компонентов ($1 \leqslant k \leqslant 300$) и списка этих компонентов. Каждый из компонентов сока описывается словом длиной до 30 символов из строчных и прописных букв латинского алфавита. Прописные и строчные буквы различаются. Различные компоненты имеют различные названия.

Формат выходных данных

В выходной файл выведите минимальное количество раз, которое Степану придется помыть емкость для сока. Учитывайте при этом, что емкость для сока надо помыть и после приготовления последней порции сока.

Примеры

стандартный ввод	стандартный вывод
4 1 Apple 2 Apple Orange 1 Orange 2 Orange Pineapple	2
3 1 Apple 1 Orange 1 Mango	3

Задача F. За Орду!

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сегодня великий вождь Саурфанг поведет орков в бой, и орки планируют сражаться так яростно, как никогда прежде. Любой великий вождь знает, что для успешной атаки нужно тщательное планирование.

В бою будут участвовать n орков, которых необходимо разделить на отряды. Саурфанг считает, что важны не количество или численность отрядов, а высокий боевой дух. Для каждого орка известно, что сам он пойдет в атаку с боевым духом a_i. Однако если орка воодушевит его командир, то он уже будет сражаться более яростно с боевым духом b_i. Но здесь возникает небольшая сложность...

Система подчинения орков весьма сложна. У каждого орка может быть несколько командиров, а у каждого командира может быть много подчиненных. Известно лишь, что все орки разного возраста, и никогда младший орк не может командовать старшим.

Саурфанг решил, что каждый отряд будет устроен следующим образом: вождь назначит командира отряда, который пойдет впереди. За ним будет идти его подчиненный, следующим будет подчиненный второго орка, и так далее. При этом воодушевлены будут все орки, кроме командира отряда. Возможно, что отряд будет состоять лишь из одного невоодушевленного орка.

Осталось лишь разделить орков на отряды, так, чтобы каждый орк был ровно в одном отряде, а суммарный боевой дух орков был максимальным. Но поскольку Саурфанг — орк, а орки не любят решать задачи, он пойдет точить свой топор, а сформировать отряды предстоит вам.

Формат входных данных

В первой строке задано число орков n ($1 \leqslant n \leqslant 10^5$).

Далее идет n строк. В i-й строке задано число k_i — количество орков, подчиняющихся i-му, а затем k_i чисел — номера этих орков. Орки пронумерованы в порядке старшинства, поэтому каждое число в i-й строке строго больше i. Сумма k_i не превосходит $2 \cdot 10^5$.

В следующей строке содержатся n целых чисел a_i ($1 \leqslant a_i \leqslant 10^9$).

В последней строке содержатся n целых чисел b_i ($a_i \leqslant b_i \leqslant 10^9$).

Формат выходных данных

В первой строке выведите два числа — суммарный боевой дух орков при оптимальном разбиении на отряды, и количество отрядов s.

В следующих s строках выведите описания отрядов. В начале описания выведите количество орков в отряде t_j, а затем t_j чисел — номера орков по порядку, начиная с командира отряда.

Если оптимальных разбиений существует несколько, выберите любое.

Пример

стандартный ввод	стандартный вывод
4 2 2 3 1 4 1 4 0 1 1 1 1 1 2 3 4	9 2 2 1 3 2 2 4

Задача G. Помогите воеводе (tuned version)

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Жил да был в Тридевятом царстве Никита Добрынич — главный царский воевода. Он всегда гордился своими воинами и любил приглашать Царя на строевые учения, на которых демонстрировались боевые приемы и тактики возглавляемого им отряда дружины. Долго ли — коротко ли, но однажды поссорился Никита Добрынич с Бабой Ягой (ходили слухи, что ссора произошла после того, как воевода невысоко отозвался о боевых навыках избушки на курьих ножках, чем очень расстроил Бабу Ягу).

Результатом ссоры стало довольно странное проклятье, наложенное Бабой Ягой на воеводу. Звучало оно очень умно и довольно безобидно: «Да будут все солдаты, квадрат расстояния между которыми на учениях окажется равным 5, враждовать др

Строевые учения проходят на прямоугольной площадке $n \times m$, разбитой на $n \cdot m$ квадратных участков размерами 1×1 для каждого солдата. Таким образом, квадрат расстояния между солдатами на участках (x_1, y_1) и (x_2, y_2) в точности равен $(x_1 - x_2)^2 + (y_1 - y_2)^2$, и теперь в учениях не всегда могут принимать участие сразу $n \cdot m$ солдат, как это было до проклятия Бабы Яги. Если, конечно, воевода не хочет, чтобы солдаты побили друг друга, или того хуже... Например, если поставить дружинника в квадрат $(2, 2)$, то дружинников в квадраты $(1, 4)$, $(3, 4)$, $(4, 1)$ и $(4, 3)$ поставить уже нельзя — между любым из них и дружинником в квадрате $(2, 2)$ начнет действовать проклятье Бабы Яги.

Более того, Баба Яга решила, что наложенного проклятья будет мало, и наложила порчу на некоторые k клеток: (x_1, y_1), $(x_2, y_2), \ldots,$ (x_k, y_k). Конечно же, ставить дружинников в клетки с порчей нельзя, ведь это опасно для их жизни.

Ваша задача — помочь воеводе, и по заданным размерам площадки для строевых учений сообщить, какое максимальное количество воинов можно одновременно собрать на этой площадке, чтобы ни на каких двух из них не начало действовать проклятье Бабы Яги.

Формат входных данных

В первой строке записаны два числа n и m ($1 \leqslant n, m \leqslant 300$) — размер площадки.

Во второй строке записано число k ($0 \leqslant k \leqslant n \cdot m$) — количество клеток с порчей.

В следующих k строках записаны два числа x_i и y_i ($1 \leqslant x_i \leqslant n$, $1 \leqslant y_i \leqslant m$) — координаты i-й клетки с порчей.

Гарантируется, что заданные k клеток попарно различны.

Формат выходных данных

Выведите одно число — максимальное количество воинов, которых можно безопасно собрать на площади.

Примеры

стандартный ввод	стандартный вывод
2 4 0	4
3 4 0	6
2 4 8 1 1 2 1 1 2 2 2 1 3 2 3 1 4 2 4	0
2 4 2 1 3 2 1	3

Задача H. Смешной граф

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вася называет граф *смешным*, если разность степеней любых двух вершин не превосходит 1 по абсолютной величине. У Васи уже есть некоторый *смешной* граф, и он хочет удалить несколько ребер из графа таким образом, чтобы получившийся граф также являлся *смешным*, а также степень хотя бы одной вершины полученного графа была равна d.

Помогите Васе построить такой граф.

Формат входных данных

В первой строке записаны три числа N ($1 \leqslant N \leqslant 300$) — количество вершин в графе, M ($1 \leqslant M \leqslant \frac{N(N-1)}{2}$) — количество ребер в графе и d ($0 \leqslant d \leqslant 299$) — требуемая степень.

В каждой из следующих M строк содержатся два различных числа — номера вершин, соединенных i-м ребром. Гарантируется, что в графе нет кратных ребер, а также, что граф является *смешным*.

Формат выходных данных

Если у Васи не получится построить желаемый граф, в единственной строке выведите «`NO`» (без кавычек).

В противном случае в первой строке выведите «`YES`» (без кавычек).

Во второй строке выведите количество ребер в полученном графе.

В третьей строке выведите номера ребер в полученном графе в **возрастающем** прядке. Ребра нумеруются в порядке ввода, начиная с единицы.

Если существует несколько решений, выведите любое.

Примеры

стандартный ввод	стандартный вывод
5 6 1 1 2 2 3 1 3 1 4 4 5 3 5	YES 2 2 5
4 3 3 1 2 2 3 3 4	NO

Задача I. Паросочетание vs независимое множество

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Дан неориентированный граф, состоящий из $3 \cdot n$ вершин и m ребер. Требуется найти паросочетание, состоящее из n ребер, **или** независимое множество, состоящее из n вершин.

Множество рёбер называется паросочетанием, если никакие два ребра этого множества не имеют общих вершин.

Множество вершин называется независимым, если никакие две вершины этого множества не соединены ребром.

Формат входных данных

В первой строке записано одно целое число $T \geqslant 1$ — количество графов, которое вам надо обработать. Затем следуют описания T графов.

В первой строке описания графа записаны два числа n и m, где $3 \cdot n$ — количество вершин, а m — количество ребер в графе ($1 \leqslant n \leqslant 10^5$, $0 \leqslant m \leqslant 5 \cdot 10^5$).

В следующих m строках записаны пары чисел v_i и u_i ($1 \leqslant v_i, u_i \leqslant 3 \cdot n$), означающие, что вершины v_i и u_i соединены ребром.

Гарантируется, что в графе нет петель и кратных ребер.

Гарантируется, что сумма n по всем графам в одном тесте не превышает 10^5, сумма m по всем графам в одном тесте не превышает $5 \cdot 10^5$.

Формат выходных данных

Выведите ответы для каждого из T графов. Формат ответа для одного графа описан ниже.

Если вы нашли паросочетание размера n, то в первой строке выведите «`Matching`» (без кавычек), а во второй строке выведите n чисел, разделённых пробелами, — номера рёбер паросочетания. Рёбра нумеруются от 1 до m в порядке ввода.

Если вы нашли независимое множество размера n, то в первой строке выведите «`IndSet`» (без кавычек), а во второй строке выведите n чисел, разделённых пробелами, — номера вершин независимого множества.

Если в графе нет ни того, ни другого, выведите «`Impossible`» (без кавычек).

Номера рёбер или вершин можно выводить в произвольном порядке.

Если существует несколько решений, вы можете вывести любое. В том числе, если в графе есть как паросочетание размера n, так и независимое множество размера n, то вы можете вывести любое из таких паросочетаний **или** любое из таких независимых множеств.

Пример

стандартный ввод	стандартный вывод
4 1 2 1 3 1 2 1 2 1 3 1 2 2 5 1 2 3 1 1 4 5 1 1 6 2 15 1 2 1 3 1 4 1 5 1 6 2 3 2 4 2 5 2 6 3 4 3 5 3 6 4 5 4 6 5 6	Matching 2 IndSet 1 IndSet 2 4 Matching 1 15

Замечание

Первые два графа из примера совпадают, однако в этом графе есть как паросочетание размера 1, так и независимое множество размера 1. Любое из таких паросочетаний и независимых множеств является правильным ответом.

В третьем графе нет паросочетания размера 2, однако есть независимое множество размера 2. В этом графе также есть независимое множество размера 5: `2 3 4 5 6`. Тем не менее, такой ответ не является верным, потому что вас просят найти независимое множество (или паросочетание) размера **ровно** n.

В четвёртом графе нет независимого множества размера 2, однако есть паросочетание размера 2.

Задача J. Проблема падишаха

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать максимальное число семей, причем сделать это таким образом, чтобы минимальная совместимость в семье была максимальной. А решить эту неклассическую задачу он просит вас. Помогите падишаху!

Формат входных данных

В первой строке входных данных содержатся два целых числа n и m — количество юношей и количество девушек соответственно ($1 \leqslant n, m \leqslant 200$). Последующие n строк содержат по m целых чисел от 0 до 10^9 — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

Формат выходных данных

В единственную строку выходного файла выведите наименьший искомый балл, при котором возможно создание максимально возможного количества семейных пар.

Пример

стандартный ввод	стандартный вывод
3 4 77 88 31 67 96 30 2 68 35 39 76 45	76

Задача K. Минимальное контролирующее множество

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Формат входных данных

В первой строке файла даны два числа m и n ($1 \leqslant m, n \leqslant 4\,000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leqslant K_i \leqslant n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит 500 000. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \leqslant L_i \leqslant n$ — соответствующая i-й вершине первой доли вершина второй доли, или 0, если i-я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S, после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Примеры

стандартный ввод	стандартный вывод
3 2 2 1 2 1 2 1 2 1 2 0	2 1 1 1 2