

Задача А. 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 $A \cup B = V$, $A \cap B = \emptyset$.

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 ($1 \leq n, m \leq 250$), 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.

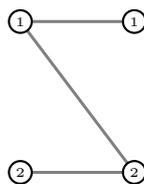
Формат выходных данных

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 соответственно.

Пример

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

Замечание



Задача С. 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 ($1 \leq n, m \leq 100, |a| \leq 1000, |b| \leq 1000$). 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. 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 свежавыжатый 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, \dots, a_n , а сок B – из компонентов b_1, \dots, 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 ($1 \leq N \leq 300$). Каждая из последующих N строк описывает один из соков. Описание сока состоит из числа k его компонентов ($1 \leq k \leq 300$) 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
<pre>4 1 Apple 2 Apple Orange 1 Orange 2 Orange Pineapple</pre>	<pre>2</pre>
<pre>3 1 Apple 1 Orange 1 Mango</pre>	<pre>3</pre>

Задача Е. 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 Тридевятиом 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 друг с другом».

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) , $(x_2, y_2), \dots, (x_k, y_k)$. 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 ($1 \leq n, m \leq 300$) — размер площадки.

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

В следующих k строках записаны два числа x_i и y_i ($1 \leq x_i \leq n$, $1 \leq y_i \leq m$) — 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 площадке.

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 F. 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 ($1 \leq n, m \leq 200$). 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 G. 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 ($1 \leq N \leq 300$) — количество вершин в графе, M ($1 \leq M \leq \frac{N(N-1)}{2}$) — количество ребер в графе и d ($0 \leq d \leq 299$) — 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** порядке. 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 Н. 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 ($1 \leq m, n \leq 4000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leq K_i \leq n$) — 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 $0 \leq L_i \leq n$ — 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
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача I. Максимизировать mex

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

В колледже учатся n студентов, также в колледже есть m клубов, пронумерованных от 1 до m . У каждого студента известен его потенциал p_i и номер клуба c_i , членом которого он является. Изначально каждый студент является членом ровно одного клуба. Скоро в колледже состоится технический фестиваль, который продлится d дней. Каждый день в рамках фестиваля будет проведено соревнование по программированию.

Каждый день утром ровно один студент решает покинуть свой клуб. После того как студент покинул свой клуб, он больше не присоединится ни к какому клубу снова. Каждый день в полдень директор колледжа выбирает по одному студенту из каждого клуба (в случае если в каком-то клубе нет ни одного студента, из этого клуба не будет выбран никто) и составляет из них команду на этот день. Силой команды называется mex потенциал студентов, которые в неё входят. Директор хочет выяснить наибольшую возможную силу команды в каждый из следующих d дней. Таким образом, каждый день директор выбирает команду так, чтобы максимизировать силу команды. Для мультимножества S его mex определён как наименьший неотрицательный элемент, не входящий в S . Например,

$$\text{mex}\{0, 1, 1, 2, 4, 5, 9\} = 3, \quad \text{mex}\{1, 2, 3\} = 0, \quad \text{mex}\emptyset = 0.$$

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq m \leq n \leq 5000$) — количество студентов и количество клубов в колледже.

Вторая строка содержит n целых чисел p_1, p_2, \dots, p_n ($0 \leq p_i < 5000$) — где p_i это потенциал i -го студента.

Третья строка содержит n целых чисел c_1, c_2, \dots, c_n ($1 \leq c_i \leq m$), обозначающие, что i -й студент изначально является членом клуба c_i .

Четвёртая строка содержит одно целое число d ($1 \leq d \leq n$) — количество дней, в течение которых продлится фестиваль.

Каждая из следующих d строк содержит одно целое число k_i ($1 \leq k_i \leq n$), обозначающее, что k_i -й студент покинул свой клуб на i -й день. Гарантируется, что к моменту каждого выхода соответствующий студент ещё состоит в клубе.

Формат выходных данных

Для каждого из d дней выведите одно целое число — наибольшую возможную силу команды в этот день.

Примеры

стандартный ввод	стандартный вывод
5 3 0 1 2 2 0 1 2 2 3 2 5 3 2 4 5 1	3 1 1 1 0
5 3 0 1 2 2 1 1 3 2 3 2 5 4 2 3 5 1	3 2 2 1 0
5 5 0 1 2 4 5 1 2 3 4 5 4 2 3 5 4	1 1 1 1

Задача J. Есть M стульев: на какой сам сядешь, на какой...

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

Есть M стульев на вещественной прямой, i -й стул ($1 \leq i \leq M$) стоит в координате i . N людей хотят сесть на стулья, но у каждого есть свои предпочтения. А именно, i -й человек может сесть только на стул с координатой не более L_i или не менее R_i . Конечно же, на одном стуле может сидеть не более одного человека.

Возможно, прямо сейчас вам не удастся рассадить всех людей по стульям согласно их требованиям. Поэтому вы можете доставить произвольное количество стульев в произвольные точки вещественной оси (не обязательно целые). Разные стулья должны быть в разных координатах (в том числе на уже поставленные M координат ставить стулья нельзя). Найдите минимальное число стульев, которое придётся доставить, чтобы посадить всех N людей.

Формат входных данных

В первой строке даны два числа N и M , ($1 \leq N, M \leq 2 \cdot 10^5$)

В i -й из следующих N строк указаны предпочтения i -го человека L_i и R_i , ($0 \leq L_i < R_i \leq M + 1, 1 \leq i \leq N$). Гарантируется, что все L_i, R_i целые.

Формат выходных данных

Выведите минимальное число дополнительных стульев, чтобы можно было посадить всех людей.

Примеры

стандартный ввод	стандартный вывод
4 4 0 3 2 3 1 3 3 4	0
7 6 0 7 1 5 3 6 2 7 1 6 2 6 3 7	2

Замечание

В первом примере 1-й, 2-й, 3-й и 4-й человек могут сесть на стулья в координатах 3, 2, 1 и 4 соответственно, и дополнительные стулья не понадобятся.

Во втором примере можно поставить один стул в координату 0 и второй в координату 2.5. После этого люди с первого по седьмого могут сесть на стулья с координатами 0, 5, 3, 2, 6, 1, 2.5 соответственно.