

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.

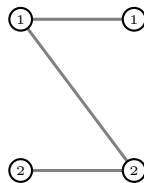
Формат выходных данных

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 В. 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 С. Разбиение на пары

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

Космические археологи обнаружили на планете в соседней звездной системе n древних артефактов, которые они пронумеровали от 1 до n . Каждый артефакт имеет k различных параметров, каждый параметр характеризуется целым числом. Артефакт i имеет параметры $a_{i,1}, a_{i,2}, \dots, a_{i,k}$. Оказалось, что первые параметры у всех артефактов различны: для всех $i \neq j$ выполнено $a_{i,1} \neq a_{j,1}$, при этом другие параметры у артефактов могут совпадать.

Учёные также обнаружили текст, в соответствии с которым для активации артефактов их необходимо особым образом разбить на пары и совместить. Разбиение артефактов на пары является корректным, если для каждого t от 1 до k можно выбрать такое число b_t , что оно лежит на отрезке между значениями t -го параметра артефактов каждой пары. То есть, если артефакты i и j образуют пару, должно выполняться условие $a_{i,t} \leq b_t \leq a_{j,t}$ или условие $a_{i,t} \geq b_t \geq a_{j,t}$.

Теперь ученые хотят выяснить, верно ли расшифрован текст. Для этого необходимо проверить, существует ли корректное разбиение артефактов на пары. Каждый артефакт должен войти ровно в одну пару в разбиении.

Требуется написать программу, которая по описанию параметров артефактов определяет, можно ли разбить их на пары таким образом, чтобы для каждого параметра существовало значение, лежащее между значениями этого параметра артефактов каждой пары, и в случае положительного ответа выводит такое разбиение.

Формат входных данных

В первой строке заданы целые числа n и k – количество артефактов и количество параметров ($2 \leq n \leq 2 \cdot 400$, n чётно, $1 \leq k \leq 7$).

В следующих n строках задано по k целых чисел $a_{i,1}, a_{i,2}, \dots, a_{i,k}$ – параметры артефактов ($-10^9 \leq a_{i,j} \leq 10^9$, все значения $a_{i,1}$ различны).

Формат выходных данных

Выведите «NO», если требуемого разбиения на пары не существует.

В противном случае выведите «YES» в первой строке. Далее выведите $n/2$ строк, в каждой строке выведите по два числа – номера артефактов, из которых следует составить пару. Каждый артефакт должен быть выведен ровно один раз.

Если существует несколько корректных разбиений артефактов на пары, разрешается вывести любое из них

Примеры

стандартный ввод	стандартный вывод
6 2 8 6 1 5 6 3 3 1 4 7 7 2	YES 4 1 5 6 2 3
4 3 1 -1 -1 2 1 1 3 -1 1 4 1 -1	NO

Задача D. 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 Е. 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 — совсем 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 минимизировать 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 F. 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 ($2 \leq n \leq 1000$, $0 \leq m \leq 10^5$). 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 G. 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
4 1 Apple 2 Apple Orange 1 Orange 2 Orange Pineapple	2
3 1 Apple 1 Orange 1 Mango	3

Задача Н. 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 ($1 \leq n \leq 10^5$).

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 ($1 \leq a_i \leq 10^9$).

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

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	9 2
2 2 3	2 1 3
1 4	2 2 4
1 4	
0	
1 1 1 1	
1 2 3 4	

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