

Задача А. Паросочетание

Имя входного файла: `pairs.in`
Имя выходного файла: `pairs.out`
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

<code>pairs.in</code>	<code>pairs.out</code>
2 2	2
1 2 0	1 1
2 0	2 2

Задача В. Минимальное контролирующее множество

Имя входного файла: `minimal.in`
Имя выходного файла: `minimal.out`
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

<code>minimal.in</code>	<code>minimal.out</code>
3 2	2
2 1 2	1 1
1 2	1 2
1 2	
1 2 0	

Задача С. День рождения

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

Митя знаком с m юношами и n девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов k ($1 \leq k \leq 20$). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа $0 \leq m \leq 150$ и $0 \leq n \leq 150$. Далее следуют m строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком i -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Если максимальных наборов несколько, то выведите любой из них.

Примеры

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

Задача D. Паросочетание максимального веса

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

Дан двудольный граф. Количество вершин в левой и правой доле совпадает и равно n . У каждой вершины левой доли есть вес, i -й вершине соответствует вес w_i . Вес паросочетания, ребрам которого инцидентны вершины левой доли a_1, a_2, \dots, a_k есть $\sqrt{\sum_{i=1}^k w_{a_i}^2}$. Требуется найти паросочетание максимального веса.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество вершин в обеих долях ($1 \leq n \leq 1000$). Вторая строка входного файла содержит n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 1000$). Следующие n строк содержат описания ребер, инцидентных соответствующей вершине левой доли. Формат описания: количество ребер, затем номера вершин правой доли, разделенные пробелом. Суммарное количество ребер не превосходит 200000.

Формат выходных данных

Выведите n чисел — для каждой вершины левой доли выведите номер вершины правой доли, с которой ее надо взять в паросочетание. Если вершина не входит в паросочетание, выведите 0.

Примеры

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

Задача Е. Такси

Имя входного файла: `taxi.in`
Имя выходного файла: `taxi.out`
Ограничение по времени: 0.5 секунда
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

<code>taxi.in</code>	<code>taxi.out</code>
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. Толстые хоббиты

Имя входного файла: `hobbits.in`
Имя выходного файла: `hobbits.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Ни один хоббит не в состоянии в одиночку противостоять полчищам Мордора... В последний поход против Мордора Гэндальф решил отправить N хоббитов из Шира. Но часть хоббитов наотрез отказалась, жалуясь на то, что другие хоббиты наверняка будут дразнить их толстыми. После опроса всех хоббитов оказалось, что любой хоббит отказывается принять участие в походе в том случае, если с ним в поход выступит хотя бы один хоббит с меньшим весом. К счастью для Средиземья, не все хоббиты знают свой точный вес. В Шире были всего одни весы чашечного типа, позволяющие для пары хоббитов определить, какой хоббит тяжелее. Некоторые пары хоббитов взвешивались на этих весах. Всем хоббитам известен результат всех взвешиваний. Гэндальф абсолютно уверен, что в Шире нет двух хоббитов одного веса. Он заинтересован в том, чтобы отряд состоял из наибольшего количества хоббитов. Однако найти наибольшее множество хоббитов, среди которых ни один не считает себя тяжелее другого, оказалось не так-то просто. Подскажите Гэндальфу, на сколько хоббитов он может рассчитывать. Помните при этом, что хоббиты умные существа и знают, что если Сэм тяжелее Пиппина, а Пиппин тяжелее Фродо, то Сэм и подавно будет тяжелее Фродо.

Формат входных данных

В первой строке дано целое число N – количество хоббитов ($2 \leq N \leq 100$). Все хоббиты пронумерованы целыми числами от 1 до N . В следующих N строках записана матрица размера $N \times N$. Если i -й и j -й хоббит взвешивались на чашечных весах и оказалось, что i -й хоббит тяжелее, то в i -й строке матрицы на j -й позиции стоит единица. Во всех остальных случаях в матрице стоят нули.

Формат выходных данных

В первой строке выведите размер наибольшего множества хоббитов, готового выступить в поход, во второй строке перечислите номера хоббитов из этого множества через пробел.

Примеры

<code>hobbits.in</code>	<code>hobbits.out</code>
2 0 1 0 0	1 2
3 0 0 0 0 0 0 0 0 0	3 1 2 3

Задача G. Танцевальная зарядка

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

В ЛКШ традиционно проводится турнир по хастлу. К сожалению, дети на кастинг приходят не парами, и получилось так, что количество пришедших мальчиков было ровно на два больше, чем количество пришедших девочек. Разумеется, не все пары мальчиков и девочек изъявили желание танцевать друг с другом.

Жюри турнира хочет, чтобы турнир получился *эффектным*. Для этого необходимо разбить мальчиков и девочек на разнополюе пары, таким образом, чтобы выполнить два условия:

- каждая пришедшая на кастинг девочка должна участвовать в турнире;
- мальчик и девочка танцуют вместе только если они изъявили такое желание.

Было решено отсеять на кастинге ровно двух мальчиков, чтобы количество участвующих мальчиков было равно количеству участвующих девочек. Жюри хочет знать количество пар мальчиков, таких что если они не пройдут кастинг, можно будет провести эффектный турнир.

Формат входных данных

Первая строка входных данных содержит единственное число n ($1 \leq n \leq 10\,000$) — количество девочек, пришедших на кастинг. В i -й из следующих $n+2$ строк описаны девочки, с которыми может танцевать i -й мальчик (то есть они оба согласны встать в пару). Сначала идёт число l_i ($0 \leq l_i \leq n$) — количество различных девочек, с которыми может танцевать i -й мальчик. Далее следует l_i чисел от 1 до n — номера этих девочек.

Гарантируется, что суммарное количество возможных пар m (сумма всех l_i) не превышает 20 000.

Формат выходных данных

Выведите количество различных пар мальчиков, непрохождение кастинга которыми позволит провести эффектный турнир.

Примеры

<code>dance.in</code>	<code>dance.out</code>
3 0 2 1 2 2 2 3 0 2 2 3	1
3 1 1 0 2 1 3 0 1 2	1

Задача Н. Работающие лягушки

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

До появления Великого Уквакнителя все лягушки делились на два типа. Первые говорили «ква», а вторые – «куа». В это тяжелое время всем n лягушкам приходилось работать на болоте для процветания их водоема. Известно, что лягушка i работала с l_i часа до r_i часа включительно.

Из-за разногласий в произношении лягушки могли работать только если в их «офисе» не было лягушки, которая говорит не так, как она.

Ваша задача состоит в том, чтобы помочь лягушкам и выбрать из них максимальное количество так, чтобы одновременно на работе были только лягушки, которые говорят одинаково, то есть не должно быть пар лягушек, что их время работы пересекается и одна из них говорит «ква», а другая «куа».

Формат входных данных

В первой строке задано единственное целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество лягушек.

В последующих n строках задано по три целых числа l_i, r_i, t_i ($1 \leq l_i \leq r_i \leq 10^9; t_i \in \{1, 2\}$) — описание i -ой лягушки (ее время работы и тип речи — «ква» или «куа»)

Формат выходных данных

Выведите максимальное количество лягушек, которые можно выбрать так, чтобы среди выбранных лягушек не было пары, которая не может вместе работать.

Примеры

стандартный ввод	стандартный вывод
2 1 2 1 2 3 2	1
3 1 3 1 4 6 2 2 5 1	2
5 5 8 1 1 3 2 3 4 2 6 6 1 2 10 2	4