

Задача А. Планирование заданий

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

Имеется некоторое множество заданий и один исполнитель. На выполнение одного задания уходит единица времени. Задания можно выполнять начиная с момента времени 0. У каждого задания есть две характеристики: d_i и w_i . Если задание не было выполнено к моменту времени d_i , взимается штраф в размере w_i . Требуется минимизировать суммарный штраф.

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество заданий ($1 \leq n \leq 100000$). Следующие n строк содержат по два натуральных числа, разделенных пробелом — d_i и w_i ($0 \leq d_i, w_i \leq 10^9$).

Формат выходных данных

Выведите одно число — минимальный суммарный штраф.

Пример

стандартный ввод	стандартный вывод
2 1 1 1 2	1

Задача В. 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.

Пример

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

Задача D. Ремонт дорог

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

Славная страна Матроидландия состоит из N городов, соединенных M шоссе. Каждое шоссе соединяет два различных города, и по каждому шоссе можно перемещаться в обе стороны. Некоторые пары городов могут быть соединены более чем одной дорогой. Правительство Матроидландии решило произвести модернизацию дорожной системы в стране. В стране есть несколько компаний, занимающихся ремонтом дорог. Они уже договорились, кто что будет ремонтировать, таким образом, для каждого шоссе указана компания, которая будет его ремонтировать.

Чтобы поддержать малый бизнес в развивающейся стране, Правительство решило, что каждая компания может ремонтировать не более, чем одно шоссе. Так как жители Матроидландии не очень любят думать, между любыми двумя городами должен существовать максимум один путь по отремонтированным дорогам. Ваша задача состоит в том, чтобы определить максимальное число шоссе, которые можно отремонтировать, чтобы все были довольны.

Формат входных данных

Первая строка входного файла содержит два числа — N и M ($1 \leq N \leq 100$, $0 \leq M \leq 5000$). Следующие M строк содержат описания шоссе. Каждое описание состоит из двух номеров городов u и v , которые соединяет шоссе, и номера c компании, которая это шоссе собирается ремонтировать ($1 \leq v, u \leq N$, $1 \leq c \leq 200$).

Формат выходных данных

На первой строке выведите единственное число K — искомое максимальное количество шоссе. На следующей строке выведите номера этих шоссе в любом порядке. Если существует несколько решений, выведите любое.

Примеры

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

Задача Е. Два остовных дерева

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

Задан неориентированный граф, рёбра которого можно разбить на два непересекающихся остовных дерева. Вам необходимо найти одно из таких разбиений.

Формат входных данных

Первая строка входного файла содержит два натуральных числа N ($N \leq 400$) и M ($M \leq 700$) — количество вершин и рёбер в графе.

Следующие M строк содержат описания рёбер графа. Каждое ребро задается номерами концов. Гарантируется, что в графе нет петель, но могут быть кратные рёбра. Вершины и рёбра графа нумеруются с единицы.

Формат выходных данных

Выходной файл должен содержать искомое разбиение рёбер графа.

В первой строке выведите номера рёбер, которые войдут в первое остовное дерево, во второй — номера рёбер, которые войдут во второе остовное дерево. Каждое ребро графа должно появиться ровно в одной из этих двух строк.

Пример

стандартный ввод	стандартный вывод
6 10	3 4 5 6 8
1 2	1 2 7 9 10
2 3	
3 4	
4 5	
5 6	
6 1	
6 2	
2 5	
5 3	
3 4	

Задача F. Дорожные вопросы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Короли Берляндии и Флатландии в очередной раз запланировали дорожные реформы в своих государствах. На этот раз они решили действовать сообща.

Берляндия и Флатландия — два крупных государства, в каждом из которых есть n городов, между которыми правители хотят построить дороги. В результате тендера короли выбрали m подрядчиков, i -й из которых построит двустороннюю дорогу между городами a_i и b_i в Берляндии и двустороннюю дорогу между городами c_i и d_i во Флатландии, в результате чего суммарное благополучие этих стран возрастёт на w_i . Обратите внимание, что w_i может быть отрицательным, так как не все подрядчики добросовестно выполняют свою работу. В случае если подрядчик будет нанят, он построит обе дороги, нельзя попросить его построить только одну из них.

Короли Берляндии и Флатландии очень заботятся об экономичности транспортных систем своих стран, а именно они никогда не допустят, чтобы между какими-то двумя городами в их странах существовало два различных простых пути между этими городами. Путь называется простым, если он посещает каждый город не более одного раза, два пути называются различными, если различаются множества дорог, которые эти пути используют. Обратите внимание, что короли могут построить дорожную систему, в которой между некоторыми двумя городами нет пути по дорогам. Иными словами, короли хотят, чтобы графы, образованные дорогами каждой страны, были лесами — множествами неориентированных деревьев.

Короли ещё не решили, сколько же дорог они построят, поэтому просят вас для каждого k от 1 до m определить, какого максимального суммарного благополучия стран они могут добиться, если наймут k подрядчиков, при условии, что дорожные сети каждой страны должны получиться экономичными.

Формат входных данных

В первой строке задано два целых числа n и m ($2 \leq n \leq 800$, $1 \leq m \leq 800$) — количество городов в каждой из стран и число подрядчиков, соответственно.

В следующих m строках заданы описания подрядчиков.

Описание i -го подрядчика задано пятью целыми числами a_i , b_i , c_i , d_i , и w_i ($1 \leq a_i, b_i, c_i, d_i \leq n$, $a_i \neq b_i$, $c_i \neq d_i$, $-10^9 \leq w_i \leq 10^9$) — номера городов в Берляндии, которые соединит дорога i -го подрядчика, номера городов во Флатландии, которые соединит дорога i -го подрядчика и величина, на которую возрастёт суммарное благополучие стран, если дороги i -го подрядчика будут построены.

Формат выходных данных

Выведите m строк, в i -й строке выведите одно целое число — максимально возможное суммарное благополучие стран, если будут наняты **ровно** i подрядчиков или «Impossible» (без кавычек), если нельзя получить экономичную дорожную сеть, наняв i подрядчиков.

Примеры

стандартный ввод	стандартный вывод
4 3 1 2 1 2 7 1 3 2 1 8 2 3 3 2 6	8 14 Impossible
6 4 1 2 1 3 34 2 3 3 2 11 2 4 3 1 5 2 1 3 5 8	34 45 24 Impossible
3 2 3 1 2 3 -9 2 3 1 3 -21	-9 -30

Замечание

Рассмотрим первый пример.

При $k = 1$ выгодно нанять подрядчика с номером 2. При $k = 2$ выгодно нанять подрядчиков с номерами 2 и 3. Единственный способ нанять трёх подрядчиков — это нанять всех подрядчиков, однако получившаяся дорожная сеть не будет экономичной так как во Флатландии будет существовать два простых пути между городами 1 и 2 (по дороге, построенной первым подрядчиком, и по дороге, построенной вторым подрядчиком).

В третьем примере все возможные множества подрядчиков построят эффективные дорожные сети.

Задача G. Рюкзак-матроид

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

Одна из версий задачи о рюкзаке формулируется следующим образом: дано n предметов, i -й из них имеет вес w_i и стоимость v_i , требуется выбрать множество предметов с суммарным весом, не превышающим c (емкости рюкзака) и максимальной возможной суммарной стоимостью. Известно, что задача о рюкзаке является NP-полной. Для нее существуют решения с помощью динамического программирования, но они обычно имеют время работы, линейное относительно суммы весов предметов и емкости рюкзака, которое, таким образом, не является полиномиальным относительно размера входных данных.

Однако бывают случаи, когда задачу о рюкзаке можно так или иначе решить жадно. Один важный пример такого класса задач — если набор весов таков, что множество решений задачи образует матроид.

Матроидом называется пара $\langle X, \mathcal{I} \rangle$, где X — конечное множество, а \mathcal{I} семейство подмножеств X , которые называют независимыми. При этом \mathcal{I} должно удовлетворять следующим трем свойствам:

1. $\mathcal{I} \neq \emptyset$;
2. Если $A \in \mathcal{I}$ и $B \subset A$, то $B \in \mathcal{I}$;
3. Если $A, B \in \mathcal{I}$ и $|A| > |B|$, то найдется такой $x \in A \setminus B$, что $B \cup \{x\} \in \mathcal{I}$.

Например, ребра неориентированного графа и семейство их ациклических подмножеств образуют матроид.

Рассмотрим предметы с весами w_1, w_2, \dots, w_n . Пусть X представляет собой множество целых чисел от 1 до n . Будем называть подмножество $\{i_1, i_2, \dots, i_k\}$ независимым, если $w_{i_1} + w_{i_2} + \dots + w_{i_k} \leq c$. Проверьте, образует ли получившаяся пара матроид.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 50$). Вторая строка содержит n целых чисел w_1, w_2, \dots, w_n ($1 \leq w_i \leq 100$). Третья строка содержит число c ($\min w_i \leq c \leq \sum w_i$).

Формат выходных данных

Выведите «YES», если множество решений задачи о рюкзаке образует матроид. В противном случае выведите «NO».

Во втором случае выведите на второй строке число 2 или 3 — номер свойства, которое нарушается. Следующие две строки должны содержать контрпример к указанному свойству. Первая из строк должна описывать множество A , а вторая — B . Описание множества должно состоять из числа элементов в множестве и затем списка входящих в него предметов.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 4	YES
3 3 4 5 7	NO 3 2 1 2 1 3

Задача Н. Я что, похож на абонента?

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	256 мегабайт

Дана сетка $n \times n$, состоящая из квадратиков 1×1 . В центре каждого квадратика 1×1 стоит телефонная вышка. В некоторых вершинах сетки (т. е. в углах квадратиков) стоят абоненты. Каждая вышка может соединиться с какими-то из абонентов, стоящих в углах ее квадратика, у каждой вышки есть ограничение на количество абонентов, с которыми она может соединиться. Абонент может соединиться с любым количеством вышек из соседних к нему.

Вам нужно для каждой вышки выбрать тех абонентов, которые подсоединятся к ней так, чтобы минимизировать количество компонент связности.

Более формально, компонента связности — это максимальное по включению множество абонентов и вышек, что для любой пары из них существует путь из одного в другого по связям вышка-абонент.

Формат входных данных

Первая строка входных данных содержит единственное целое число n ($1 \leq n \leq 100$) — размер сетки.

Каждая из следующих $4 * n + 1$ строк содержит по $4 * n + 1$ символов — описание сетки. Каждая клетка сетки задается в виде множества из 3×3 блоков, заполненных символом «.». Если в центре клетки находится вышка, то в центре множества 3×3 содержится цифра от 1 до 4 — ограничение на количество соединений вышки.

Стороны сетки заданы с помощью символа «*». В вершинах сетки содержится либо символ «А», либо «*», символ «А» обозначает, что вершине находится абонент, «*» — иначе.

Формат выходных данных

В первой строке выведите единственное целое число — минимальное количество компонент связности.

В следующих $4 * n + 1$ строках выведите по $4 * n + 1$ символов — описание полученное сетки. Формат описания сетки в ответе совпадает с описанием формата во входных данных. Соединения обозначаются символом «\» (если соединение идет от вышки к абоненту в левом-верхнем или правом-нижнем углу множества из 3×3 блоков) или «/» (если соединение идет от вышки к абоненту в правом-верхнем или левом-нижнем углу множества из 3×3 блоков).

Примеры

стандартный ввод	стандартный вывод
<pre> 3 ***** *...*...*...* *...*.2.*...* *...*...*...* ****A***A**** *...*...*...* *.2.*...*.2.* *...*...*...* ****A***A**** *...*...*...* *...*.2.*...* *...*...*...* ***** </pre>	<pre> 1 ***** *...*...*...* *...*.2.*...* *...*/.*...* ****A***A**** *../*...*\...* *.2.*...*.2.* *..*...*/...* ****A***A**** *...*\./...* *...*.2.*...* *...*...*...* ***** </pre>
<pre> 3 A***A***A***A *...*...*...* *.3.*.2.*.3.* *...*...*...* A***A***A***A *...*...*...* *.1.*...*.2.* *...*...*...* A***A***A***A *...*...*...* *.4.*.2.*.3.* *...*...*...* A***A***A***A </pre>	<pre> 4 A***A***A***A *\./*\./*\./* *.3.*.2.*.3.* *..*...*/...* A***A***A***A *...*...*.../* *.1.*...*.2.* *..*...*...* A***A***A***A *\./*...*\...* *.4.*.2.*.3.* */.*/.*/.* A***A***A***A </pre>
<pre> 3 ****A***A***A *...*...*...* *.2.*.2.*.2.* *...*...*...* ****A***A***A *...*...*...* *.2.*.2.*.2.* *...*...*...* ****A***A***A *...*...*...* *.3.*.3.*.2.* *...*...*...* A***A***A**** </pre>	<pre> 1 ****A***A***A *../*../*../* *.2.*.2.*.2.* *..*...*...* ****A***A***A *../*../*../* *.2.*.2.*.2.* *..*...*...* ****A***A***A *../*../*../* *.3.*.3.*.2.* */.*/.*/...* A***A***A**** </pre>