

Содержание

Задача 1A. Проверка [2 секунды, 64 мегабайта]	2
Задача 1B. Паросочетание максимального веса [2 seconds, 256 megabytes]	3
Задача 1C. Реер Blue [2 секунды, 256 мегабайт]	4
Задача 1D. Ремонт дорог [2 секунды, 256 мегабайт]	5
Задача 1E. Два остовных дерева [2 секунды, 256 мегабайт]	6
Задача 1F. Milk Candy [2 секунды, 512 мегабайт]	8

Задача 1А. Проверка [2 секунды, 64 мегабайта]

Дано некоторое семейство множеств $S \subset 2^X$. Требуется проверить, может ли S быть семейством независимых множеств некоторого матроида.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m — мощность множеств X и S соответственно ($1 \leq n \leq 10$, $0 \leq m \leq 2^n$). Каждая из следующих m строк содержит описание элемента множества S . Формат описания: количество элементов в подмножестве, затем через пробел номера этих элементов. Элементы множества X занумерованы, начиная с единицы.

Формат выходных данных

Выведите «YES», если S может быть семейством независимых множеств некоторого матроида, и «NO» иначе.

Примеры

стандартный ввод	стандартный вывод
2 4 0 1 1 1 2 2 1 2	YES
2 3 0 1 1 2 1 2	NO

Задача 1В. Паросочетание максимального веса [2 seconds, 256 megabytes]

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

standard input	standard output
4	2 1 0 4
1 3 2 4	
4 1 2 3 4	
2 1 4	
2 1 4	
2 1 4	

Задача 1С. Peep Blue [2 секунды, 256 мегабайт]

Суперкомпьютер Peep Blue *умеет* играть в игры. Однако Серёжа не побоялся принять его вызов.

Правила, предложенные компьютером, таковы. Есть n кучек камней. Игроки ходят по очереди, начинает Серёжа. На первом ходу каждому из игроков разрешается убрать любой набор кучек (в том числе пустой, но не все). На втором и следующих ходах каждый из игроков может взять любое ненулевое число камней из любой кучки. Проигрывает тот, кто не может сделать ход.

Ваша задача — определить, может ли Серёжа одержать победу, а также какой первый ход ему следует сделать. Среди всех первых ходов, приносящих победу, выберите такой, который оставит в игре как можно больше камней: это должно усложнить задачу Peep Blue.

Формат входных данных

В первой строке ввода записано целое число n — количество кучек камней ($1 \leq n \leq 100\,000$). В следующей строке записаны n натуральных чисел, a_i , не превышающих $2^{30} - 1$, — количества камней в кучках.

Формат выходных данных

Если Серёжа не может одержать победу, выведите единственное число -1 . Иначе выведите максимально возможное суммарное число камней, которое можно оставить.

Примеры

peep.in	peep.out
4 1 2 3 4	9

Задача 1D. Ремонт дорог [2 секунды, 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

highways.in	highways.out
4 5	3
1 2 1	1 4 5
3 1 1	
2 3 1	
1 4 2	
3 4 3	

Задача 1Е. Два остовных дерева [2 секунды, 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 ($N \leq 600$) и M — количество вершин и ребер в графе. Следующие M строк содержат описания ребер графа. Каждое ребро задается номерами концов. Гарантируется, что в графе нет петель, но могут быть кратные ребра. 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.

Примеры

twost.in	twost.out
6 10 1 2 2 3 3 4 4 5 5 6 6 1 6 2 2 5 5 3 3 4	3 4 5 6 8 1 2 7 9 10
18 34 4 13 3 5 11 12 16 7 9 3 15 2 13 9 6 2 12 10 11 1 5 4 8 12 10 9 6 10 13 18 1 4 2 6 14 16 2 4 17 11 10 3 5 3 17 5 9 12 1 3 14 8 18 5 4 12 8 13 16 14 18 3 7 15 15 17 7 11	1 2 3 4 5 6 7 8 9 12 13 14 16 20 27 30 32 10 11 15 17 18 19 21 22 23 24 25 26 28 29 31 33 34

Задача 1F. Milk Candy [2 секунды, 512 мегабайт]

Калабаш играет в RPG игру на своём компьютере. В этой игре есть n неизвестных чисел x_1, x_2, \dots, x_n и m NPC продающих NPC. i -й NPC продаёт c_i подсказок. Каждая подсказка выглядит как (l_j, r_j, w_j) , что означает что стоимость подсказки равна w_j , и что эта подсказка сообщит Калабашу значение $x_{l_j} + x_{l_j+1} + \dots + x_{r_j-1} + x_{r_j}$.

Чтобы пройти игру нужно выяснить все n неизвестных чисел. Хитрый Калабаш знает как покупать хинты независимо, но NPC хитрее. Калабаш обязан купить у i -го NPC ровно k_i подсказок. Конечно, одну подсказку нельзя покупать несколько раз.

Эта задача уже слишком трудна для Калабаша. Вычислите минимальное количество денег, которое вам понадобится, или определите, что пройти игру нельзя.

Формат входных данных

В первой строке записаны n и m ($1 \leq n, m \leq 80$), количество неизвестных чисел и количество NPC.

Далее следует m описаний NPC. Каждое описание начинается со строки из двух целых чисел c_i и k_i ($1 \leq k_i \leq c_i$), обозначающих общее количество подсказок и количество подсказок, которые нужно купить.

Каждая из следующих c_i строк содержит целые числа l_j, r_j и w_j ($1 \leq l_j \leq r_j \leq n$, $1 \leq w_j \leq 10^6$), описывающее подсказки i -го NPC.

Гарантируется, что сумма c_i в одном тесте не превосходит 80.

Формат выходных данных

Выведите минимальное количество денег. Если решения нет, выведите “-1”.

Пример

стандартный ввод	стандартный вывод
2 2 1 1 1 2 1 3 2 1 1 10 2 2 100 1 2 1000	111
2 2 1 1 1 1 1 1 1 1 1 2	-1