

Задача 1. МЕХ подмножеств

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

Мультимножество представляет собой набор элементов, в котором элементы могут повторяться несколько раз. Например, следующий набор является мультимножеством:

$\{0, 0, 1, 2, 2, 5, 5, 5, 8\}$

Дано мультимножество S содержащее неотрицательные целые числа, и целевое неотрицательное число n , такое что n не лежит в S . Ваша цель состоит в том, чтобы добавить n в S , используя следующие трех шаговые операции несколько раз:

1. Выберите (возможно пустое) подмножество T из S . Здесь T — это множество **различных** чисел, все элементы которого входят в S .
2. Удалите элементы T из S . (Удалите только одну копию каждого элемента.)
3. Добавьте $\text{mex}(T)$ в S , где $\text{mex}(T)$ — наименьшее неотрицательное целое число, не принадлежащее T . Название mex происходит от английского «minimum excluded» value (минимальное отсутствующее число).

Ваша цель — найти минимальное количество операций, которые нужно выполнить, чтобы n стало частью S .

Поскольку размер S может быть большим, мультимножество будет задано в виде списка $(f_0, \dots, f_{n-1})$ размера n , где f_i указывает сколько раз число i появляется в S . (Напомним, что n — это целое число, которое мы пытаемся добавить в S .)

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leq t \leq 200$) — количество наборов входных данных.

Первая строка каждого набора входных данных содержит одно целое число n ($1 \leq n \leq 50$) — число, которое нужно добавить в S .

Вторая строка каждого набора входных данных содержит n целых чисел $f_0, f_1, \dots, f_{n-1}$ ($0 \leq f_i \leq 10^{16}$), описывающие мультимножество S как указано выше.

Формат выходных данных

Для каждого набора входных данных выведите одну строку, содержащую минимальное количество операций, необходимых для удовлетворения условия задачи.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	5	$n \leq 2$		первая ошибка
2	17	$n \leq 20$	1	первая ошибка
3	7	$f_i = 0$		первая ошибка
4	9	$f_i \leq 1$	3	первая ошибка
5	20	$f_i \leq 2000$	3, 4	первая ошибка
6	9	$f_j = 0$ для $j \neq 0$		первая ошибка
7	10	существует значение i , для которого $f_j = 0, j \neq i$	6	первая ошибка
8	23	нет	1 — 7	первая ошибка

Пример

стандартный ввод	стандартный вывод
2 4 0 3 0 3 5 4 1 0 2 0	4 10

Замечание

В первом примере изначально $S = \{1, 1, 1, 3, 3, 3\}$ и наша цель добавить 4 в S . Мы можем сделать следующее:

1. выбрать $T = \{\}$, затем S станет $\{0, 1, 1, 1, 3, 3, 3\}$
2. выбрать $T = \{0, 1, 3\}$, затем S станет $\{1, 1, 2, 3, 3\}$
3. выбрать $T = \{1\}$, затем S станет $\{0, 1, 2, 3, 3\}$
4. выбрать $T = \{0, 1, 2, 3\}$, тогда S станет $\{3, 4\}$

Задача 2. Путешествие

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	1024 мегабайта

Петя отправляется в кругосветное путешествие. Он определил для себя количество стран n , через которые он планирует проехать. Для удобства пронумеруем страны целыми числами от 1 до n . Изначально Петя находится в стране с номером 1, а его пунктом назначения является страна с номером n . Между странами существует m односторонних КПП, каждый из которых соединяет две различные страны.

Петя подготовился к путешествию и выяснил для каждого КПП, сколько времени он на нем проведет. Однако, из-за снегопада эти значения могли значительно измениться. Теперь для каждого КПП известно целое число t_i — количество дополнительного времени (иными словами, задержка), которое придется прождать на КПП, либо количество сэкономленного времени, если $t_i < 0$.

Кроме того, в некоторых странах продают специальные пропуски, с помощью которых Петя может встать в специальную очередь к КПП, в которой время ожидания не изменилось относительно того, каким оно было до снегопада (то есть, можно считать, что в этой очереди $t_i = 0$). Один пропуск можно использовать только один раз, но Петя может иметь при себе несколько пропусков. Он получает по одному пропуску каждый раз, когда посещает страну, где их продают. Из-за странных законов Пете запрещено иметь с собой более, чем n пропусков.

Маршрут Пети должен начинаться со страны с номером 1 и заканчиваться в стране с номером n . Перемещаться между странами можно только через КПП (возможно, используя специальные очереди при наличии пропусков). Петя не хочет, чтобы путешествие затягивалось, поэтому маршрут должен включать не более, чем $2 \cdot m$ КПП. Помогите ему найти маршрут, удовлетворяющий данным ограничениям, минимизирующий суммарную задержку по времени, проведенному на КПП.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n \leq 80$, $1 \leq m \leq 500$) — количество стран и КПП, соответственно.

Вторая строка содержит n целых чисел $s_1, s_2, \dots, s_n$ ($s_i \in \{0, 1\}$) — если $s_i = 1$, то в i -й стране продают пропуски, а иначе — нет.

Каждая из следующих m строк содержит три целых числа a_i , b_i и t_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq |t_i| \leq 100\,000$) — это значит, что существует КПП, позволяющий проехать из страны a_i в страну b_i . Задержка на этом КПП составляет t_i единиц времени.

Формат выходных данных

Сначала выведите два целых числа t и l ($1 \leq l \leq 2 \cdot m$) — минимальную суммарную задержку, а также количество КПП, через которые проходит маршрут Пети.

В каждой из следующих l строк выведите два целых числа w_i и p_i ($1 \leq w_i \leq m$, $p_i \in \{0, 1\}$). w_i означает номер КПП, через который должен проехать Петя. $p_i = 0$, если Петя должен использовать пропуск, и $p_i = 1$ в противном случае.

Если существует несколько оптимальных маршрутов с разным количеством КПП, вы можете вывести любой из них.

Гарантируется, что существует хотя бы один маршрут.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

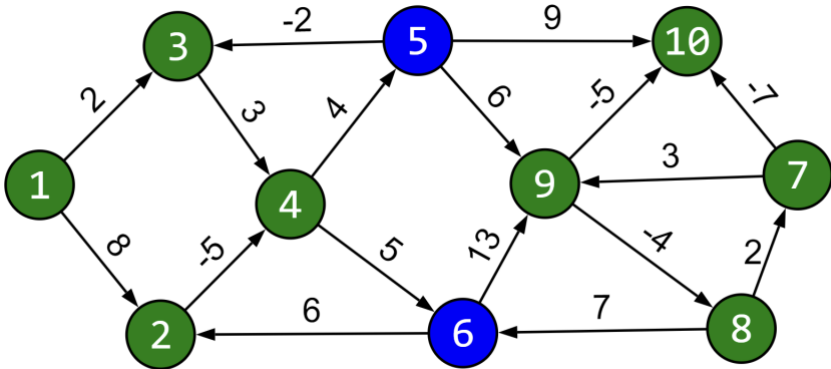
Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	10	$n \leq 10, m \leq 50, s_i = 0, t_i \geq 1$		первая ошибка
2	25	$s_i = 0, t_i \geq 1$	1	первая ошибка
3	25	$t_i \geq 1$	1, 2	первая ошибка
4	25	$s_i = 0$	1, 2	первая ошибка
5	15	нет	1 – 4	первая ошибка

Пример

стандартный ввод	стандартный вывод
10 17	-2 7
0 0 0 0 1 1 0 0 0 0	2 1
1 3 2	3 1
1 2 8	6 1
2 4 -5	9 0
4 6 5	11 1
6 2 6	13 1
4 5 4	14 1
5 3 -2	
3 4 3	
5 9 6	
6 9 13	
9 8 -4	
8 6 7	
8 7 2	
7 10 -7	
9 10 -5	
7 9 3	
5 10 9	

Замечание

На изображении ниже показаны страны и КПП из первого примера. Синим обозначены страны, в которых продаются пропуска. Оптимальный маршрут Пети выглядит следующим образом: $1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 9 \rightarrow 8 \rightarrow 7 \rightarrow 10$. Пропуск, полученный в стране 5, использован на КПП при перемещении из страны 5 в страну 9.



Задача 3. Раскраска

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

Вам дан массив из n целых чисел $a_1, a_2, \dots, a_n$. От вас требуется ответить на q независимых запросов.

Каждый запрос характеризуется двумя целыми числами x_i и y_i . Раскрасим все числа в массиве, которые равны x_i или y_i . Пусть раскрашенными оказались элементы с номерами $p_1, p_2, \dots, p_k$. Ваша цель — минимизировать значение суммы $\sum_{i=1}^k \sum_{j=1}^k |p_i - p_j|$. Для достижения этой цели за одну операцию вы можете поменять местами два любых соседних элемента массива. Вычислите минимальное количество операций, при выполнении которых данная сумма будет минимально возможной.

Формат входных данных

Первая строка содержит два целых числа n и q ($2 \leq n \leq 10^6$, $1 \leq q \leq 10^6$) — количество элементов в массиве и количество запросов, соответственно.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) — элементы массива.

Каждая из следующих q строк содержит два целых числа x_i и y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$) — параметры i -го запроса.

Гарантируется, что среди элементов массива a есть хотя бы два различных числа.

Формат выходных данных

Для каждого запроса выведите одно целое число — минимальное количество операций, необходимых, чтобы минимизировать указанную сумму.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	15	$n, q \leq 15$		первая ошибка
2	13	$n, q \leq 500$	1	первая ошибка
3	13	$n, q \leq 10\,000$	1, 2	первая ошибка
4	29	$n, q \leq 200\,000$	1, 2, 3	первая ошибка
5	30	нет	1 – 4	первая ошибка

Примеры

стандартный ввод	стандартный вывод
5 4 1 2 3 1 4 1 2 1 3 1 4 2 4	1 1 2 2
10 6 5 2 1 3 1 3 3 2 1 2 1 2 2 3 3 1 1 5 5 2 3 5	8 4 1 6 11 4

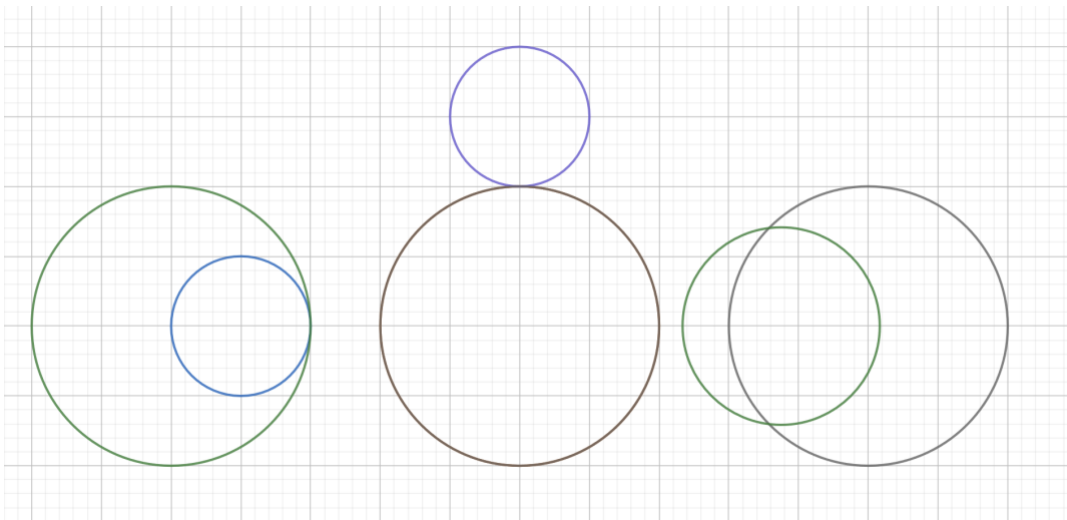
Задача 4. Похищение

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

Маленькую овечку Бе (сокращенно от Бериллий) похитили пришельцы, и у них к ней есть довольно необычная просьба. Они хотят нанять ее на работу.

Сегодня пришельцы планируют посетить Землю на n космических кораблях и наградить лучших участников олимпиады по дрон-рейсингу (и, возможно, тоже нанять их). Их космические корабли представляют собой идеальные круги.

В целях безопасности они выбрали m пар космических кораблей, которые должны касаться внешним образом посадке. Они уже определили координаты посадки центральной точки для каждого из космических кораблей, и задача Бе состоит в определении радиуса каждого из космических кораблей, чтобы удовлетворять условиям задачи.



*На изображении выше левая и правая пара кораблей **не** удовлетворяют условию внешнего касания. Пара кораблей по центру удовлетворяет условию, так как окружности касаются внешним образом.*

Космические корабли достаточно дорогие в производстве, и их цена равна площади соответствующего круга. Поэтому пришельцы попросили Бе определить радиусы окружностей таким образом, чтобы минимизировать суммарную стоимость кораблей.

Технологии пришельцев позволяют кораблям пересекаться при посадке. Более того, они умеют строить корабли с радиусом, равным нулю.

Если выбрать радиусы требуемым образом невозможно, Бе должна это выяснить. Если Бе не сможет определить радиусы или выяснить, что это невозможно, её наймут... на обед. Помогите овечке не попасть на эту не привлекательную вакансию.

Формат входных данных

Первая строка содержит два целых числа n и m ($1 \leq n, m \leq 100\,000$) — количество космических кораблей, а также количество пар кораблей, которые должны касаться внешним образом при посадке.

Каждая из следующих n строк содержит два **вещественных** числа x_i и y_i ($-10\,000 \leq x_i, y_i \leq 10\,000$) — координаты центра i -го корабля при посадке. Все числа заданы с 10-ю знаками после запятой.

Каждая из следующих m строк содержит два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — номера кораблей, окружности которых должны касаться внешним образом. Каждая неупорядоченная пара кораблей (a, b) встречается во входных данных не более одного раза.

Формат выходных данных

В случае, если выбрать радиусы невозможно, в единственной строке выведите «NE».

В противном случае в первой строке выведите «DA».

Далее выведите n вещественных чисел $r_1, r_2, \dots, r_n$ — радиусы кораблей.

В случае, если существует несколько оптимальных ответов, выведите любой из них.

Ответ будет считаться верным, если абсолютная или относительная погрешность суммы площадей кораблей не превосходит 10^{-4} .

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
0	0	Тесты из условия		полная
1	15	n чётно, каждый корабль должен касаться ровно двух других кораблей		первая ошибка
2	25	Гарантируется, что выбрать радиусы требуемым образом возможно		первая ошибка
3	30	Для каждой пары космических кораблей (a, b) существует не более одной последовательности различных космических кораблей, начинающейся с корабля a и заканчивающейся кораблем b , такой что соседние космические корабли в последовательности должны касаться		первая ошибка
4	30	нет	1, 2, 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 3 0.0000000000 0.0000000000 0.0000000000 2.0000000000 2.0000000000 0.0000000000 1 2 2 3 3 1	DA 0.585786 1.414214 1.414214
5 4 -0.4585133080 0.2893567973 9.9368007273 7.1806641913 -8.4621834970 -2.8309311865 0.0122121945 -2.8309311865 2.3991780589 -8.8626906628 2 1 3 2 4 3 5 1	DA 0.000000 12.472076 8.474396 0.000000 9.587824
5 5 0.0000000000 0.0000000000 1.0000000000 2.0000000000 2.0000000000 4.0000000000 3.0000000000 6.0000000000 4.0000000000 8.0000000000 1 2 2 3 3 4 4 5 5 1	NE

Замечание

В первом примере существует единственный подходящий набор радиусов (с точностью до погрешности).

В третьем примере выбрать радиусы требуемым образом невозможно.