

## Задача А. Алгоритм двух китайцев

Имя входного файла: `chinese.in`  
Имя выходного файла: `chinese.out`  
Ограничение по времени: 6 секунд  
Ограничение по памяти: 256 мегабайт

Вам дан взвешенный ориентированный граф, содержащий  $n$  вершин и  $m$  рёбер. Найдите минимально возможную сумму весов  $n - 1$  ребра, которые нужно оставить в графе, чтобы из вершины с номером 1 по этим ребрам можно было добраться до любой другой вершины.

### Формат входных данных

В первой строке даны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 1000$ ,  $0 \leq m \leq 10000$ ) — количество вершин и ребер в графе.

В следующих  $m$  строках даны ребра графа. Ребро описывается тройкой чисел  $a_i$ ,  $b_i$  и  $w_i$  ( $1 \leq a_i, b_i \leq n$ ;  $-10^9 \leq w_i \leq 10^9$ ) — номер вершины, из которой исходит ребро, номер вершины, в которую входит ребро, и вес ребра.

### Формат выходных данных

Если нельзя оставить подмножество ребер так, чтобы из вершины с номером 1 можно было добраться до любой другой, в единственной строке выведите «NO».

Иначе, в первой строке выведите «YES», а во второй строке выведите минимальную возможную сумму весов ребер, которых необходимо оставить.

### Примеры

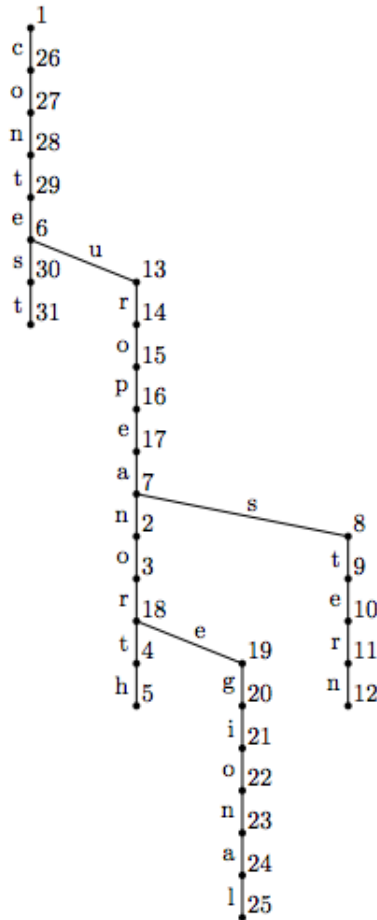
<code>chinese.in</code>	<code>chinese.out</code>
2 1 2 1 10	NO
4 5 1 2 2 1 3 3 1 4 3 2 3 2 2 4 2	YES 6

## Задача В. Словарь

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

Петя и Дима работают над новым алгоритмом сжатия данных. Их задача состоит в том, что сжать данный набор слов. Для этого они хотят построить корневое дерево, где на каждом ребре написана ровно одна буква.

Определим для такого дерева словарь, который содержит в точности те слова, которые могут быть получены конкатенацией букв на некотором пути в этом дереве (не обязательно начинающемся в корне), идущим вниз к листу (но не обязательно заканчивающимся в листе).



Ребята хотят построить такое дерево, для которого соответствующий словарь будет содержать все слова из исходного множества (и, возможно, какие-то еще слова). Среди таких деревьев они хотят выбрать то, которое содержит минимальное количество вершин. Помогите им!

На картинке выше корень дерева имеет номер 1, а путь от вершины 7 до вершины 5 соответствует слову «north», путь от вершины 16 до вершины 12 соответствует слову «eastern», путь от вершины 29 до вершины 2 соответствует слову «european», путь от вершины 3 до вершины 25 соответствует слову «regional», а путь от вершины 1 до вершины 31 соответствует слову "contest".

### Формат входных данных

Первая строка входных данных содержит число слов в множестве  $n$  ( $1 \leq n \leq 50$ ). Следующие  $n$  строк содержат различные непустые слова, по одному на строке, каждое из которых состоит из маленьких английских букв. Каждое слово состоит из не более, чем 10 символов.

## Формат выходных данных

В первой строке выведите количество вершин в исходном дереве  $m$ . В следующих  $m$  строках выведите описания вершин дерева. Вершины нумеруются с 1, описание вершины состоит из номера вершины-предка и символа, написанного на ребре, ведущего предка. Для корневой вершины описание должно состоять из единственного числа 0.

## Пример

стандартный ввод	стандартный вывод
5	31
north	0
eastern	1 c
european	2 o
regional	3 n
contest	4 o
	5 r
	6 e
	7 u
	8 r
	9 o
	10 p
	11 e
	12 a
	13 n
	13 s
	15 t
	16 e
	17 r
	18 n
	7 g
	20 i
	21 o
	22 n
	6 t
	24 h
	23 a
	26 l
	4 t
	28 e
	29 s
	30 t

## Замечание

Пример соответствует рисунку из условия.

## Задача С. Алгоритм двух китайцев. Эпизод второй

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

Вам дан взвешенный ориентированный граф, содержащий  $n$  вершин и  $m$  рёбер. Найдите минимально возможную сумму весов  $n - 1$  ребра, которые нужно оставить в графе, чтобы из вершины с номером 1 по этим ребрам можно было добраться до любой другой вершины.

### Формат входных данных

Входные данные содержат описание одного или более тестов.

Каждый тест описывается следующим образом. В первой строке даны два целых числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 300\,000$ ) — количество вершин и ребер в графе.

В следующих  $m$  строках даны ребра графа. Ребро описывается тройкой чисел  $a_i$ ,  $b_i$  и  $w_i$  ( $1 \leq a_i, b_i \leq n$ ;  $-10^5 \leq w_i \leq 10^9$ ) — номер вершины, из которой исходит ребро, номер вершины, в которую входит ребро, и вес ребра.

Сумма  $n$  по всем тестам не более 100 000. Сумма  $m$  по всем тестам не более 300 000.

### Формат выходных данных

Для каждого теста выведите или суммарный вес выбранных рёбер, или «NO», если нельзя оставить подмножество рёбер так, чтобы из вершины с номером 1 можно было добраться до любой другой.

### Пример

стандартный ввод	стандартный вывод
2 1	NO
2 1 10	6
4 5	
1 2 2	
1 3 3	
1 4 3	
2 3 2	
2 4 2	

## Задача D. Hung Fu

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

Есть два массива целых чисел длины  $n$ :  $a$  и  $b$ . Рассмотрим следующую формулу:

$$\sum_{i=1}^n \min_{1 \leq j \leq i} a_i \oplus b_j$$

Вы тренируетесь в вычислении результата данной формулы, и заметили, что порядок элементов в массивах играет значение. Теперь вы хотите минимизировать результат вычисления, переставив элементы массивов  $a$  и  $b$ . Более формально, вы хотите найти такую перестановку  $p$  которая минимизирует следующую функцию:

$$F(p) = \sum_{i=1}^n \min_{1 \leq j \leq i} a_{p_i} \oplus b_{p_j}$$

Найдите лексикографически минимальную такую перестановку  $p$ , которая минимизирует функцию.

### Формат входных данных

В первой строке вам дано одно целое число  $n$  — размер массивов ( $1 \leq n \leq 50$ ).

Во второй строке даны  $n$  целых чисел  $a_i$  — элементы массива  $a$  ( $0 \leq a_i \leq 10^6$ ).

В третьей строке даны  $n$  целых чисел  $b_i$  — элементы массива  $b$  ( $0 \leq b_i \leq 10^6$ ).

### Формат выходных данных

В первой строке выведите минимальный возможный результат функции. Во второй строке выведите  $n$  целых чисел — лексикографически минимальную перестановку  $p$ , которая минимизирует результат функции.

### Примеры

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

## Задача Е. Задача о назначениях

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана целочисленная матрица  $C$  размера  $n \times n$ . Требуется выбрать  $n$  ячеек так, чтобы в каждой строке и каждом столбце была выбрана ровно одна ячейка, а сумма значений в выбранных ячейках была минимальна.

### Формат входных данных

Первая строка входного файла содержит  $n$  ( $2 \leq n \leq 300$ ). Каждая из последующих  $n$  строк содержит по  $n$  чисел:  $C_{ij}$ . Все значения во входном файле неотрицательны и не превосходят  $10^6$ .

### Формат выходных данных

В первую строку выходного файла выведите одно число — искомая минимизируемая величина. Далее выведите  $n$  строк по два числа в каждой — номер строки и столбца клетки, участвующей в оптимальном назначении.

Пары чисел можно выводить в произвольном порядке.

### Пример

стандартный ввод	стандартный вывод
3	3
3 2 1	2 1
1 3 2	3 2
2 1 3	1 3

### Замечание

Обратите внимание, что эту задачу можно сдать с помощью алгоритма за  $O(n^3)$ , без использования Венгерского алгоритма, а используя только максимальный поток минимальной стоимости.

## Задача F. Расписание работ

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

Для защиты местной свалки от расхитителей мусора были наняты несколько охранников. Необходимо составить расписание, по которому охранники будут работать в парах, по одной паре каждую ночь. Начальник свалки нанял вас, чтобы написать программу, которая по характеристикам охранников определит максимальное возможное количество охранников, которые могут быть в расписании (остальные будут уволены). Обратите внимание, что каждый охранник может работать только в одной паре и не может работать один.

### Формат входных данных

В первой строке находится одно число  $N \leq 222$  — количество охранников. В следующих строках находится неупорядоченные пары  $(i, j)$ , каждая из которых означает, что охранники с номерами  $i$  и  $j$  могут работать вместе, так как можно найти форму, которая устроит их обоих. (На свалке используются разные части формы для разных охранников, например шлемы, штаны, куртки. Нельзя одеть маленький шлем на охранника с большой головой, или большие ботинки на охранника с маленькими ногами). Ввод заканчивается концом файла.

### Формат выходных данных

Необходимо вывести оптимальное расписание. В первой строке выведите четное число  $C$ , количество охранников в расписании. После этого выведите  $C/2$  строк, содержащих по 2 числа  $i, j$  каждая, обозначающих, что охранники  $i$  и  $j$  будут работать вместе.

### Пример

стандартный ввод	стандартный вывод
3	2
1 2	1 2
2 3	
1 3	

## Задача G. Совершенство

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Павел Андреевич очень любит все совершенное. Его новая любовь - совершенные паросочетания! Не огорчайте Павла Андреевича, проверьте, есть ли в данном графе совершенное паросочетание.

### Формат входных данных

В первой строке даны два числа  $N$  и  $M$  — количество вершин и количество ребер в  $G$ .  $N \leq 100$ . Следующие  $M$  строк содержат числа  $a_i$  и  $b_i$  — ребра графа. Гарантируется отсутствие петель и кратных ребер

### Формат выходных данных

Выведите слово «YES» или слово «NO» — ответ на поставленную задачу.

### Примеры

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



## Задача Н. Расписание работ

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.3 секунд
Ограничение по памяти:	256 мегабайт

Для защиты местной свалки от расхитителей мусора были наняты несколько охранников. Необходимо составить расписание, по которому охранники будут работать в парах, по одной паре каждую ночь. Начальник свалки нанял вас, чтобы написать программу, которая по характеристикам охранников определит максимальное возможное количество охранников, которые могут быть в расписании (остальные будут уволены). Обратите внимание, что каждый охранник может работать только в одной паре и не может работать один.

### Формат входных данных

В первой строке находится одно число  $N \leq 222$  — количество охранников. В следующих строках находится неупорядоченные пары  $(i, j)$ , каждая из которых означает, что охранники с номерами  $i$  и  $j$  могут работать вместе, так как можно найти форму, которая устроит их обоих. (На свалке используются разные части формы для разных охранников, например шлемы, штаны, куртки. Нельзя одеть маленький шлем на охранника с большой головой, или большие ботинки на охранника с маленькими ногами). Ввод заканчивается концом файла.

### Формат выходных данных

Необходимо вывести оптимальное расписание. В первой строке выведите четное число  $C$ , количество охранников в расписании. После этого выведите  $C/2$  строк, содержащих по 2 числа  $i, j$  каждая, обозначающих, что охранники  $i$  и  $j$  будут работать вместе.

### Пример

стандартный ввод	стандартный вывод
3	2
1 2	1 2
2 3	
1 3	

## Задача I. Банах

Имя входного файла: `banach.in`  
Имя выходного файла: `banach.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

Стефан, Дэвид и Феликс готовят соревнование по правилам ICPC. Стефан предлагают следующую задачу:

Дано  $N$  точек на плоскости, а также  $N$  векторов. Найдите однозначное соответствие между векторами и точками так, что если передвинуть каждую точку на соответствующий вектор, то расстояние между **любой** парой точек не уменьшится.

Дэвид справился с решением этой задачи довольно быстро и утверждает, что эта задача слишком проста для этого соревнования. Для того, чтобы убедить его, Феликс предложил следующее усложнение: среди всех возможных решений, найдите то, которое максимизирует сумму квадратов всех результирующих попарных расстояний.

Дэвиду все еще кажется, что задача слишком простая, а вам?

### Формат входных данных

Первая строка входного файла содержит число  $N$ , количество точек и векторов ( $1 \leq N \leq 500$ ).

Следующие  $N$  строк описывают точки. Каждая из них содержит два числа  $px_i$  и  $py_i$  ( $0 \leq |px_i|, |py_i| \leq 10\,000$ ).

Затем следуют  $N$  строк с описанием векторов. Каждый вектор задан двумя числами  $vx_i$  и  $vy_i$  ( $0 \leq |vx_i|, |vy_i| \leq 10\,000$ ).

### Формат выходных данных

Если способ сопоставить точки и вектора таким образом, чтобы все попарные расстояния не уменьшились существует, выведите “Yes” в первой строке выходного файла. На следующей строке, выведите  $N$  различных чисел от 1 до  $N$ ,  $i$ -е из которых равно номеру вектора, сопоставленного  $i$ -й точке.

Не забудьте выбрать ответ с максимально возможной суммой квадратов всех результирующих попарных расстояний. Если существует несколько возможных решений, выведите любое из них.

Если требование выполнить невозможно, выведите “No” в единственной строке выходного файла.

### Примеры

<code>banach.in</code>	<code>banach.out</code>
2 0 0 1 0 2 0 -2 0	Yes 2 1
2 2 2 2 2 -1 -1 -1 -1	Yes 1 2

### Замечание

В первом примере, существует только два возможных способа сопоставить векторы точкам. В обоих сопоставлениях, “1 2” и “2 1”, расстояние между каждой парой точек не уменьшается. В первом случае сумма квадратов расстояний между всеми парами точек равна 9, но во втором случае она равна 25. Поэтому единственным правильным решением является “2 1”.

Во втором примере, опять же, только два случая установить соответствие. В обоих случаях, расстояние между всеми парами точек не уменьшается, и сумма квадратов попарных расстояний равна 0. Поэтому, оба ответа корректны.

## Задача J. Баланс

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

Назовем матрицу  $A$  размера  $N \times N$  *сбалансированной*, если  $A[i][j] + A[i + 1][j + 1] = A[i + 1][j] + A[i][j + 1]$  для всех  $1 \leq i, j \leq N - 1$ . Вам дана матрица  $A$  размера  $N \times N$ . Ваша задача вывести другую матрицу  $B$  того же размера, что  $B$  сбалансированна и  $B[i][j] \geq A[i][j]$  для всех  $1 \leq i, j \leq N$ . Более того, матрица  $B$  должна иметь минимально возможную сумму всех элементов.

### Формат входных данных

Первая строка содержит число  $N$ , число строк и столбцов в матрице ( $1 \leq N \leq 50$ ). Каждая из следующих  $N$  строк содержит  $N$  чисел. Вместе они описывают матрицу  $A$ . Гарантируется, что  $0 \leq A[i][j] \leq 35\,000$  для всех  $1 \leq i, j \leq N$ .

### Формат выходных данных

В первой строке выведите сумму значений в сбалансированной матрице  $B$ . В следующих  $N$  строках, выведите сбалансированную матрицу в том же формате, как и во входных данных.

Значения в выведенной матрице ничем не ограничены (например, они могут превосходить 35 000).

### Пример

стандартный ввод	стандартный вывод
4	16
1 1 1 1	1 1 1 1
1 1 1 1	1 1 1 1
1 1 1 0	1 1 1 1
1 1 1 1	1 1 1 1

## Задача К. Стандартная задача на графы

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

Вам дан взвешенный ориентированный граф, содержащий  $n$  вершин и  $m$  ребер. Каждая вершина графа может быть выделенной или обычной, изначально все вершины обычные. Ценой графа назовём минимальную суммарную цену рёбер, которые нужно взять, чтобы из каждой обычной вершины была достижима по взятым ребрам хотя бы одна любая выделенная вершина. Если такого набора рёбер не существует, то цена равна  $-1$ .

Вам предстоит вычислить цену графа после каждого из  $q$  запросов. Запросы бывают двух типов:

- $+ v_i$  делает вершину  $v_i$  выделенной; гарантируется, что перед запросом вершина была обычной.
- $- v_i$  делает вершину  $v_i$  обычной; гарантируется, что перед запросом вершина была выделенной.

Выведите цену графа после каждого из  $q$  запросов.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $m$  и  $q$  ( $3 \leq n \leq 2 \cdot 10^5$ ,  $1 \leq m, q \leq 2 \cdot 10^5$ ) — количество вершин графа, количество рёбер и число запросов.

Следующие  $m$  строк содержат рёбра графа, по одному ребру в каждой строке.  $i$ -я строка содержит три целых числа  $u_i$ ,  $v_i$  и  $c_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $1 \leq c_i \leq 10^6$ ) — концы  $i$ -го ребра (из  $u_i$  в  $v_i$ ), а также его вес ( $c_i$ ).

Следующие  $q$  строк содержат запросы, по одному запросу в каждой строке.  $i$ -я строка содержит  $+ v_i$ , если это операция первого типа, и  $- v_i$ , если это операция второго типа ( $1 \leq v_i \leq n$ ).

### Формат выходных данных

Выведите  $q$  чисел.  $i$ -е число — это цена графа после первых  $i$  запросов.

## Примеры

стандартный ввод	стандартный вывод
4 5 6 1 2 1 2 3 5 3 2 3 4 1 8 2 1 4 + 1 - 1 + 3 + 1 + 4 + 2	15 -1 14 12 4 0
10 14 10 8 6 4 2 5 1 3 5 4 1 6 3 1 3 7 7 2 1 6 1 3 4 10 1 4 6 5 5 4 1 5 8 10 10 9 1 9 5 1 9 7 6 + 7 + 8 - 7 + 10 + 2 - 10 + 5 - 2 - 5 + 3	28 24 29 19 18 24 18 19 29 20

## Замечание

В первом тесте:

- После первого запроса выгоднее всего взять ребра с номерами 3, 4, 5, сумма их цен равна 15.
- После второго запроса, нет ни одной выделенной вершины, а значит, не существует подходящих наборов ребер, цена графа  $-1$ .
- После третьего запроса выгоднее всего взять ребра с номерами 1, 2, 5, сумма их цен равна 14.
- После четвертого запроса выгоднее всего взять ребра с номерами 4 и 5, сумма их цен равна 12.
- После пятого запроса выгоднее всего взять только ребро номер 5, его цена равна 4.

- После шестого запроса все вершины выделенные и можно не брать ребер, цена графа равна 0.