

ИТМО. АиСД. y2022. М3234-3237. Третий семестр

10А. Суффиксный массив

2 s., 256 MB

Постройте суффиксный массив для заданной строки *s*.

Входные данные

Первая строка входного файла содержит строку *s* ( $1 \leq |s| \leq 400\,000$ ). Строка состоит из строчных латинских букв.

Выходные данные

Выведите  $|s|$  различных чисел — номера первых символов суффиксов строки *s* так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

входные данные
ababb
выходные данные
1 3 5 2 4

Так как учитель знает, что Александр Г. и Илья С. очень любят списывать у известного в узких кругах Демида Г., каждый школьник получил своё число *k* и вынужден был обратиться к вам за помощью.

Входные данные

В первой строке входного файла находится строка *S* ( $|S| \leq 10^5$ ). Вторая строка содержит число *k* ( $1 \leq k \leq 10^{18}$ ) — порядковый номер запрашиваемой подстроки.

Выходные данные

Если ответ существует, выведите искомую подстроку строки *S*. В противном случае выведите её лексикографически максимальную подстроку.

входные данные
abacaba 10
выходные данные
acab

10В. Циклические сдвиги

2 секунды, 256 мегабайт

*k*-м циклическим сдвигом строки *S* называется строка, полученная перестановкой *k* первых символов строки *S* в конец строки.

Рассмотрим все различные циклические сдвиги строки *S* и отсортируем их по возрастанию. Требуется вычислить *i*-ю строчку этого массива.

Например, для строки *abacabac* существует четыре различных циклических сдвига: нулевой (*abacabac*), первый (*bacabaca*), второй (*acabacab*) и третий (*cabacaba*). После сортировки по возрастанию получится такой массив: *abacabac*, *acabacab*, *bacabaca*, *cabacaba*.

Входные данные

В первой строке входного файла записана строка *S*, длиной не более 100 000 символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число *k* ( $1 \leq k \leq 100\,000$ ).

Выходные данные

В выходной файл выведите *k*-й по возрастанию циклический сдвиг строки *S*, или слово IMPOSSIBLE, если такого сдвига не существует.

входные данные
abacabac 4
выходные данные
cabacaba

10С. Контрольное списывание

1 s., 256 MB

Сегодня на уроке преподаватель Массивов Автомат Укконевич рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки *A* и *B* лексикографически. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконевич дал им следующее задание: найти *k*-ю лексикографически непустую уникальную подстроку строки *S*.

10D. Рефрен

2 секунды, 256 мегабайт

Рассмотрим последовательность *n* целых чисел от 1 до *m*. Подпоследовательность подряд идущих чисел называется рефреном, если произведение её длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти её рефрен.

Входные данные

Первая строка входного файла содержит два целых числа: *n* и *m* ( $1 \leq n \leq 150\,000, 1 \leq m \leq 10$ ).

Вторая строка содержит *n* целых чисел от 1 до *m*.

Выходные данные

Первая строка выходного файла должна содержать произведение длины рефрена на количество её вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

входные данные
9 3 1 2 1 2 1 3 1 2 1
выходные данные
9 9 1 2 1 2 1 3 1 2 1

10Е. Палиндромы

0.5 секунд, 256 мегабайт

Вам дана строка *s* из маленьких английских букв. Определим для подстроки *красоту* как количество вхождений подстроки в строку, умноженное на длину подстроки. Для данной строки найдите максимальную красоту среди всех её палиндромных подстрок.

Входные данные

Первая строка содержит строку *s*. Длина строки не превосходит  $3 \cdot 10^5$ .

Выходные данные

Выведите одно число — ответ на задачу.

входные данные
abacaba

выходные данные
7
входные данные
www
выходные данные
4

входные данные
0101010101
выходные данные
18 010101

Условие  
недоступно  
на  
русском  
языке

10F. Анализ битовых строк

1 секунда, 256 мегабайт

Вы думаете, что анализировать битовые строки просто? Только не когда вы делаете это во сне!

Итак, вы спите. Внезапно, правда? Но... Боюсь, что это не тот сон, в котором вы хотели бы оказаться. В вашем сне есть битовая строка, очень длинная строка, и вам предстоит с ней что-то делать. Вам вполне ясно, что нужно проснуться как можно быстрее, и для этого вам придется найти самую лучшую *специальную* строку.

К счастью, вы помните, что вчера читали книгу по теории *специальных* строк. Единственное, что вы смогли вспомнить, это весьма странное определение *специальной* строки, которое изложено ниже.

Предположим, что у вас есть битовая строка  $T$  длины  $n$ . Обозначим биты числа как  $T_1, T_2, \dots, T_n$ . Пусть  $A(i, j)$  и  $B(i, j)$  — это количество нулевых и единичных битов среди  $T_i, T_{i+1}, \dots, T_j$  соответственно. Строк  $T$  называется *специальной* тогда и только тогда, когда для любого  $i$  от 1 до  $n$ , включительно, выполнены оба условия:  $A(1, i) \geq B(1, i)$  и  $A(i, n) \leq B(i, n)$ .

Но вы не будете рады любой *специальной* строке. Вы хотите найти наилучшую! Как вы уже поняли, сон довольно странный, поэтому правила, по которым определяется, какая строка является лучшей, тоже весьма необычные. Пусть  $L_1$  и  $L_2$  — длины двух строк, а  $P_1$  и  $P_2$  — сколько раз эти строки встречаются в данной вам строке  $S$  в качестве подстроки, соответственно. Тогда вы считаете, что первая строка лучше второй, если  $L_1 \cdot P_1 > L_2 \cdot P_2$ .

Итак, ваша задача проста... или нет? Нужно найти наилучшую *специальную* строку.

**Входные данные**  
В единственной строке записана битовая строка  $S$  ( $2 \leq |S| \leq 2 \cdot 10^5$ ).

**Выходные данные**  
В первой строке выведите число  $L \cdot P$ , где  $L$  — длина найденной лучшей *специальной* строки, а  $P$  — количество раз, которое она встречается в строке  $S$  как подстрока.

Во второй строке выведите саму найденную строку. Если есть несколько наилучших *специальных* строк, выведите любую.

Гарантируется, что в строке  $S$  есть хотя бы одна *специальная* подстрока.

входные данные
00111001110101
выходные данные
8 01

входные данные
00011001110101
выходные данные
14 00011001110101

11B. Сычи и ACM

5 секунд, 256 мегабайт

Есть  $n$  сычей (программистов). Известны пары друзей. Три сыча образуют команду на ACM, если все три попарно дружат. Требуется найти количество команд сычей на ACM.

**Входные данные**  
Первая строка входных данных содержит два целых числа  $n$  и  $m$  — количество сычей и дружеских связей соответственно ( $1 \leq n, m \leq 3 \cdot 10^5$ ). Каждая из следующих  $m$  строк содержит по два целых числа от 1 до  $n$  — номера сычей, которые дружат. Гарантируется, что никакая пара не указана два раза и сыч не дружит сам с собой.

**Выходные данные**  
В единственной строке выходного файла выведите количество компаний сычей.

входные данные
6 6 1 2 2 3 3 1 4 2 3 4 5 1
выходные данные
2

11C. Мощные юнги

0.8 секунд, 256 мегабайт

Имеется список из  $n$  юнг, для каждого из которых известен его рост  $a_1, a_2, \dots, a_n$ . Рассмотрим некоторый его подсписок  $a_l, a_{l+1}, \dots, a_r$ , где  $1 \leq l \leq r \leq n$ , и для каждого натурального числа  $s$  обозначим через  $K_s$  число юнг с ростом  $s$  в этом подписке. Назовем *мощностью* подписка сумму произведений  $K_s \cdot K_s \cdot s$  по всем различным натуральным  $s$ . Так как количество различных чисел в массиве конечно, сумма содержит лишь конечное число ненулевых слагаемых.

Необходимо вычислить мощности каждого из  $t$  заданных подписков.

**Входные данные**  
Первая строка содержит два целых числа  $n$  и  $t$  ( $1 \leq n, t \leq 200000$ ) — длина списка и количество запросов соответственно.

Вторая строка содержит  $n$  натуральных чисел  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — рост юнг.

Следующие  $t$  строк содержат по два натуральных числа  $l$  и  $r$  ( $1 \leq l \leq r \leq n$ ) — индексы левого и правого концов соответствующего подписка.

**Выходные данные**

Выведите  $t$  строк, где  $i$ -ая строка содержит единственное натуральное число — мощность подписка  $i$ -го запроса.

входные данные
3 2 1 2 1 1 2 1 3
выходные данные
3 6

входные данные
8 3 1 1 2 2 1 3 1 1 2 7 1 6 2 7
выходные данные
20 20 20

11D. Различные числа

1 секунда, 256 мегабайт

Сколько различных чисел на отрезке массива?

**Входные данные**  
На первой строке длина массива  $n$  ( $1 \leq n \leq 300\,000$ ). На второй строке  $n$  целых чисел от 0 до  $10^9 - 1$ . На третьей строке количество запросов  $q$  ( $1 \leq q \leq 300\,000$ ). Следующие  $q$  строк содержат описание запросов, по одному на строке. Каждый запрос задаётся парой целых чисел  $l, r$  ( $1 \leq l \leq r \leq n$ ).

**Выходные данные**  
Выведите ответы на запросы по одному в строке.

входные данные
5 1 1 2 1 3 3 1 5 2 4 3 5
выходные данные
3 2 3

11E. Варенье

2 секунды, 256 мегабайт

Малыш и Карлсон решили пойти на прогулку. Они знают, что прогулка будет совсем скучной, если перед ней не опустошить несколько банок варенья.

Малыш достал из кладовки  $N$  банок варенья и выставил их в ряд. В банке номер  $i$  содержится ровно  $a_i$  грамм варенья. Карлсон немного подумал и решил, что в некоторых банках недостаточно варенья, и что в банке номер  $i$  должно быть хотя бы  $b_i$  грамм варенья.

Выходить из этой ситуации Карлсон хочет в  $M$  этапов. На каждом этапе он выбирает числа  $l, r, x$  и  $y$  а затем выполняет следующие операции: в банку номер  $l$  он добавляет  $x$  грамм варенья, в банку номер  $l + 1$  —  $x + y$  грамм варенья, в банку номер  $l + 2$  —  $x + y \cdot 2$ , и так далее. В банку номер  $r$  наш герой добавит  $x + y \cdot (r - l)$  грамм варенья.

Задачи - Codeforces

Малышу хочется определить для каждой банки  $i$  наименьший номер операции, после которой в ней станет хотя бы  $b_i$  грамм варенья. Помогите Малышу: найдите соответствующее число для каждой банки.

**Входные данные**  
В первой строке входного файла задано одно число  $N$  ( $1 \leq n \leq 10^5$ ) — количество банок. Во второй строке заданы  $N$  чисел  $a_i$  ( $0 \leq a_i \leq 2 \cdot 10^9$ ) — изначальное количество варенья в банке номер  $i$ . В третьей строке заданы  $N$  чисел  $b_i$  ( $0 \leq b_i \leq 2 \cdot 10^9$ ) — минимальное количество варенья, которое должно быть в банке номер  $i$ .

В четвертой строке задано  $M$  ( $0 \leq M \leq 10^5$ ) — число этапов добавления варенья в банки, которые выполнит Карлсон. В следующих  $M$  строках описаны сами этапы в хронологическом порядке. Каждый этап задан четырьмя числами  $l, r, x$  и  $y$  ( $1 \leq l \leq r \leq N, 0 \leq x, y \leq 3 \cdot 10^5$ ).

**Выходные данные**  
Выведите  $N$  чисел в одной строке, разделенные пробелом. Число номер  $i$  должно быть равно нулю, если в банке номер  $i$  изначально было достаточно варенья, номеру этапа, после которого в ней станет хотя бы  $b_i$  варенья, или  $-1$ , если даже после выполнения всех этапов, в этой банке будет недостаточно варенья. Этапы нумеруются с единицы.

входные данные
5 5 4 4 2 1 7 7 4 7 7 3 1 2 2 0 2 5 1 1 3 4 2 2
выходные данные
1 2 0 3 -1

11F. Жесть

15 секунд, 256 мегабайт

Дам массив из  $N$  чисел. Нужно уметь обрабатывать 3 типа запросов:

- `get (L, R, x)` — сказать, сколько элементов отрезка массива  $[L..R]$  не меньше  $x$ .
- `set (L, R, x)` — присвоить всем элементам массива на отрезке  $[L..R]$  значение  $x$ .
- `reverse (L, R)` — перевернуть отрезок массива  $[L..R]$ .

**Входные данные**  
Число  $N$  ( $1 \leq N \leq 5 \cdot 10^5$ ) и массив из  $N$  чисел. Далее число запросов  $M$  ( $1 \leq M \leq 5 \cdot 10^5$ ) и  $M$  запросов. Формат описания запросов предлагается понять из примера. Для всех отрезков верно  $1 \leq L \leq R \leq N$ . Исходные числа в массиве и числа  $x$  в запросах — целые от 0 до  $10^9$ .

**Выходные данные**  
Для каждого запроса типа `get` нужно вывести ответ.

входные данные
5 1 2 3 4 5 6 get 1 5 3 set 2 4 2 get 1 5 3 reverse 1 2 get 2 5 2 get 1 1 2

выходные данные
3
1
3
1

11G. Машинное обучение

4 секунды, 512 мегабайт

На курсе машинного обучения вам выдали первое домашнее задание — вам предстоит проанализировать некоторый массив из  $n$  чисел.

В частности, вы интересуетесь так называемой *равномерностью* массива. Предположим, что в массиве число  $b_1$  встречается  $k_1$  раз,  $b_2$  —  $k_2$  раз, и т.д. Тогда *равномерностью* массива называется такое минимальное целое число  $c \geq 1$ , что  $c \neq k_i$  для любого  $i$ .

В рамках вашего исследования вы хотите последовательно проделать  $q$  операций.

- Операция  $t_i = 1, l_i, r_i$  задаёт запрос исследования. Необходимо вывести равномерность массива, состоящего из элементов на позициях от  $l_i$  до  $r_i$  включительно.
- Операция  $t_i = 2, p_i, x_i$  задаёт запрос уточнения данных. Начиная с этого момента времени  $p_i$ -му элементу массива присваивается значения  $x_i$ .

Входные данные

Первая строка содержит  $n$  и  $q$  ( $1 \leq n, q \leq 100\,000$ ) — размер массива и число запросов соответственно.

Во второй строке записаны ровно  $n$  чисел —  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ).

Каждая из оставшихся  $q$  строк задаёт очередной запрос.

Запрос первого типа задаётся тремя числами  $t_i = 1, l_i, r_i$ , где  $1 \leq l_i \leq r_i \leq n$  — границы соответствующего отрезка.

Запрос второго типа задаётся тремя числами  $t_i = 2, p_i, x_i$ , где  $1 \leq p_i \leq n$  — позиция в которой нужно заменить число, а  $1 \leq x_i \leq 10^9$  — его новое значение

Выходные данные

Для каждого запроса первого типа выведите одно число — равномерность соответствующего отрезка массива.

входные данные
10 4
1 2 3 1 1 2 2 2 9 9
1 1 1
1 2 8
2 7 1
1 2 8
выходные данные
2
3
2

Первый запрос состоит из ровно одного элемента — 1. Минимальное подходящее  $c = 2$ .

Отрезок второго запроса состоит из четырёх 2, одной 3 и двух 1. Минимальное подходящее  $c = 3$ .

Отрезок четвёртого запроса состоит из трёх 1, трёх 2 и одной 3. Минимальное подходящее  $c = 2$ .

1A. Получи дерево

1 секунда, 256 мегабайт

Задачи - Codeforces

Александр решил посадить у себя в саду дерево. Он купил его в интернет-магазине, и на следующий день его обещали доставить. Но произошла какая-то ошибка, и вместо дерева Александр получил неориентированный связный граф. Оставлять в таком виде граф нельзя: в нем могут быть циклы, в том числе петли, кратные ребра — совсем некрасиво. Поэтому Александр взял топор и решил удалить некоторое количество ребер из графа.

Выведите любое дерево, которое могло получиться у Александра в результате удаления ребер из графа.

Входные данные

Сначала вводятся 2 числа  $n$  и  $m$  — число вершин и ребер в графе ( $1 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5$ ). Далее идут  $m$  пар чисел, задающих ребра. Гарантируется, что граф связный.

Выходные данные

Выведите  $n - 1$  пару чисел — ребра, которые войдут в дерево. Ребра, как и концы каждого ребра, можно выводить в любом порядке.

входные данные
4 5
1 2
2 3
3 4
1 4
1 2
выходные данные
1 2
2 3
3 4

1B. Компоненты связности

1 секунда, 256 мегабайт

Дан неориентированный граф. Необходимо посчитать количество его компонент связности и вывести их.

Входные данные

В первой строке записаны два числа  $n$  и  $m$  ( $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$ ) — количество вершин и ребер в графе, соответственно.

Далее следуют  $m$  строк, содержащих описание ребер графа. В каждой строке записаны два числа  $u$  и  $v$  ( $1 \leq u, v \leq n, u \neq v$ ), означающие, что вершины  $u$  и  $v$  соединены ребром.

Выходные данные

В первой строке выведите одно число — количество компонент связности.

Далее выведите сами компоненты в следующем формате: в первой строке следует вывести количество вершин в компоненте, а во второй строке — номера вершин, принадлежащих данной компоненте.

входные данные
6 4
3 1
1 2
5 4
2 3
выходные данные
3
3
1 3 2
2
4 5
1
6

1C. Свяжите графа

1 секунда, 256 мегабайт

Дан неориентированный граф. Требуется добавить минимальное количество ребер, чтобы он стал связным.

**Входные данные**  
В первой строке через пробел заданы целые числа  $N$  и  $M$  — количество вершин и ребер графа ( $1 \leq N \leq 1000, 0 \leq M \leq 7000$ ). В следующих  $M$  строках через пробел заданы пары целых чисел  $u$  и  $v$  — номера вершин, соединенных ребром ( $1 \leq u, v \leq N$ ).

**Выходные данные**  
В первую строку вывести единственное число  $K$  — минимальное число рёбер, которое нужно добавить. В следующих  $K$  строках вывести по 2 числа — номера вершин, между которыми нужно провести ребро. Разрешается вывести любой ответ.

входные данные
3 1 2 1
выходные данные
1 1 3

1D. Есть ли цикл?

1 секунда, 256 мегабайт

Дан ориентированный граф. Требуется определить, есть ли в нем цикл.

**Входные данные**  
Сначала вводятся два числа:  $N$  и  $M$  ( $1 \leq N \leq 10^5, 0 \leq M \leq 10^5$ ) — количество вершин и ребер графа соответственно. Далее идет  $M$  пар чисел, задающих ребра. Гарантируется, что граф не содержит петель.

**Выходные данные**  
Выведите «-1» без кавычек, если нет цикла. В противном случае выведите в первую строку количество вершин в цикле, а во вторую — номера вершин в цикле в порядке обхода. Если ответов несколько выведите любой.

входные данные
4 4 1 2 2 3 3 4 4 1
выходные данные
4 2 3 4 1

1E. Раскраска в два цвета

1 секунда, 256 мегабайт

Вам дан неориентированный граф, состоящий из  $n$  вершин и  $m$  ребер. Требуется построить правильную раскраску вершин графа в два цвета, либо определить, что она не существует.

**Входные данные**  
Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$ ) — количество вершин и ребер графа, соответственно. Каждая из следующих  $m$  строк содержит два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ ) — номера вершин, соединенных  $i$ -м ребром графа.

Гарантируется, что граф не содержит кратных ребер.

**Выходные данные**  
В случае, если правильной раскраски вершин графа в два цвета не существует, выведите число  $-1$ .

Задачи - Codeforces

В противном случае выведите через пробел  $n$  целых чисел  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq 2$ ) — цвета вершин графа.

**Система оценки**  
Помимо тестов из условия, в данной задаче имеется 25 тестов, каждый из которых оценивается независимо в 4 балла.

входные данные
7 7 1 5 1 6 1 7 2 7 3 6 3 7 4 7
выходные данные
1 1 1 1 2 2 2

входные данные
3 3 1 2 2 3 3 1
выходные данные
-1

1F. Topsort

2 секунды, 256 мегабайт

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

**Входные данные**  
В первой строке входного файла даны два целых числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$ ) — количества вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

**Выходные данные**  
Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

входные данные
6 6 1 2 3 2 4 2 2 5 6 5 4 6
выходные данные
4 6 3 1 2 5

1G. Красно-синий граф

1 секунда, 256 мегабайт

Даны  $N$  точек, занумерованных числами  $1, 2, \dots, N$ . От каждой точки с меньшим номером к каждой точке с большим номером ведет стрелка красного или синего цвета. Раскраска стрелок называется однотонной, если нет двух таких точек  $A$  и  $B$ , что от  $A$  до  $B$  можно добраться как только по красным стрелкам, так и только по синим.

Ваша задача — по заданной раскраске определить, является ли она однотонной.

**Входные данные**  
В первой строке записано единственное число  $N$  ( $3 \leq N \leq 5\,000$ ).

В следующих  $N - 1$  строках идет описание раскраски. В  $(i + 1)$ -й строке записано  $(N - i)$  символов R (красный) или B (синий), соответствующих цвету стрелок, выходящих из точки  $i$  и входящих в точки  $(i + 1), (i + 2), \dots, N$  соответственно.

**Выходные данные**  
Выведите YES, если приведенная раскраска является однотонной, и NO в противном случае.

входные данные
3 RB R
выходные данные
NO

входные данные
3 RR R
выходные данные
YES

1Н. История о кротах

4 s., 1024 MB

Крот — маленький  
тщедушный зверек...

Примечание переводчика

Не так давно семья кротов решила прокопать новую сеть туннелей под землей. План сети, который уже разработан, состоит из подземных комнат и туннелей, которые их соединяют. Формально, план сети представляет собой неориентированный связный граф. Кротиха-мать хочет использовать данную работу как возможность научить ее двоих детей-кротов, как правильно вырыть сеть туннелей.

В качестве начальной быстрой демонстрации кротиха-мать собирается вырыть несколько комнат и туннелей, которые образуют простой путь в графе плана сети. После этого она разделит оставшиеся комнаты между своими двумя детьми так, чтобы каждому из детей досталось равное количество комнат, иначе они могут расстроиться (насчет туннелей можно не беспокоиться, потому что их копать гораздо проще). Дети могут работать в любых из своих комнат в любом порядке, в котором они захотят.

Так как у детей нет опыта по копанию сети туннелей, кротиха-мать поняла одну проблему: если между некоторой парой комнат, которые принадлежат разным детям, в плане есть туннель, то существует риск происшествия во время выкапывания данного туннеля.

Помогите кротихе-матери решить, какой путь использовать для начальной демонстрации, и как разделить оставшиеся комнаты поровну так, чтобы не существовало пары комнат, отданных разным детям, соединенных туннелем. Изначальный путь должен содержать хотя бы одну комнату и не должен посещать какую-либо комнату два раза.

**Входные данные**  
В первой строке записаны два целых числа  $c$  и  $t$  ( $1 \leq c \leq 2 \cdot 10^5$ ,  $0 \leq t \leq 2 \cdot 10^5$ ) — количество комнат и туннелей в плане.

В каждой из следующих  $t$  строк записаны два целых числа  $a$  и  $b$  ( $1 \leq a, b \leq c$ ,  $a \neq b$ ) — номера комнат, соединенных туннелем.

Комнаты пронумерованы числами от 1 до  $c$ . Между каждой парой комнат есть не более одного туннеля. Между любой парой комнат существует путь.

**Выходные данные**

Задачи - Codeforces

В первой строке выведите два целых числа  $p$  и  $s$  — количество комнат в пути, выбранном кротихой-матерью для демонстрации, и количество комнат, которые должен быть вырыть каждый из детей.

Во второй строке выведите  $p$  чисел — номера комнат, образующих путь для демонстрации. Комнаты должны быть выведены в порядке следования по пути.

В третьей строке выведите  $s$  чисел — номера комнат, отданных первому ребенку в произвольном порядке.

В четвертой строке выведите  $s$  чисел — номера комнат, отданных второму ребенку в произвольном порядке.

Гарантируется, что ответ существует. Если существует несколько корректных ответов, выведите любой.

входные данные
3 2 3 1 2 1
выходные данные
1 1 1 2 3

входные данные
4 3 1 3 2 3 3 4
выходные данные
2 1 1 3 4 2

входные данные
7 7 1 2 2 3 4 2 2 5 4 5 6 7 7 2
выходные данные
3 2 1 2 4 6 7 3 5

1I. Кратчайший путь

2 секунды, 256 мегабайт

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины  $s$  в вершину  $t$ .

**Входные данные**  
Первая строка входного файла содержит четыре целых числа  $n$ ,  $m$ ,  $s$  и  $t$  — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие  $m$  строк содержат описания дуг по одной на строке. Ребро номер  $i$  описывается тремя целыми числами  $b_i$ ,  $e_i$  и  $w_i$  — началом, концом и длиной дуги соответственно ( $1 \leq b_i, e_i \leq n$ ,  $|w_i| \leq 1000$ ).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 200\,000$ .

**Выходные данные**



Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из  $s$  в  $t$ . Если пути из  $s$  в  $t$  не существует, выведите «Unreachable».

входные данные
2 1 1 2 1 2 -10
выходные данные
-10

входные данные
2 1 2 1 1 2 -10
выходные данные
Unreachable

1J. Производство деталей

1 секунда, 256 мегабайт

Предприятие «Авто-2010» выпускает двигатели для известных во всём мире автомобилей. Двигатель состоит ровно из  $n$  деталей, пронумерованных от 1 до  $n$ , при этом деталь с номером  $i$  изготавливается за  $p_i$  секунд. Специфика предприятия «Авто-2010» заключается в том, что там одновременно может изготавливаться лишь одна деталь двигателя. Для производства некоторых деталей необходимо иметь предварительно изготовленный набор других деталей.

Генеральный директор «Авто-2010» поставил перед предприятием амбициозную задачу — за наименьшее время изготовить деталь с номером 1, чтобы представить её на выставке.

Требуется написать программу, которая по заданным зависимостям порядка производства между деталями найдёт наименьшее время, за которое можно произвести деталь с номером 1.

Входные данные

Первая строка входного файла содержит число  $n$  ( $1 \leq n \leq 10^5$ ) — количество деталей двигателя.

Вторая строка содержит  $n$  натуральных чисел  $p_1, p_2, \dots, p_n$ , определяющих время изготовления каждой детали в секундах. Время для изготовления каждой детали не превосходит  $10^9$  секунд.

Каждая из последующих  $n$  строк входного файла описывает характеристики производства деталей. Здесь  $i$ -я строка содержит число деталей  $k_i$ , которые требуются для производства детали с номером  $i$ , а также их номера. В  $i$ -й строке нет повторяющихся номеров деталей. Сумма всех чисел  $k_i$  не превосходит  $2 \cdot 10^5$ .

Известно, что не существует циклических зависимостей в производстве деталей.

Выходные данные

В первой строке выходного файла должны содержаться два числа: минимальное время (в секундах), необходимое для скорейшего производства детали с номером 1 и число  $k$  деталей, которые необходимо для этого произвести.

Во второй строке требуется вывести через пробел  $k$  чисел — номера деталей в том порядке, в котором следует их производить для скорейшего производства детали с номером 1.

входные данные
3 100 200 300 1 2 0 2 2 1

выходные данные
300 2 2 1

входные данные
2 2 3 1 2 0

выходные данные
5 2 2 1

входные данные
4 2 3 4 5 2 3 2 1 3 0 2 1 3

выходные данные
9 3 3 2 1

1K. Топологическая сортировка

1 секунда, 256 мегабайт

Задан ориентированный ациклический граф с  $n$  вершинами и  $m$  ребрами. Также задана перестановка вершин графа. Необходимо проверить, является ли данная перестановка топологической сортировкой.

Входные данные

В первой строке даны два числа  $n$  и  $m$  — количество вершин и ребер в графе соответственно ( $1 \leq n, m \leq 10^5$ ). В следующих  $m$  строках заданы пары чисел  $u_i, v_i$ , означающие, что в графе есть ребро из вершины  $u_i$  в вершину  $v_i$ . В последней строке задана перестановка из  $n$  элементов.

Выходные данные

Выведите «YES» (без кавычек), если данная перестановка является топологической сортировкой и «NO» в противном случае.

входные данные
3 3 2 3 1 3 1 2 2 1 3

выходные данные
NO

входные данные
3 3 3 2 1 2 3 1 3 1 2

выходные данные
YES

1L. Кружки в Маховниках

1 секунда, 256 мегабайт

Маленький Петя очень любит компьютеры и хочет научиться программировать.

В небольшом городке Маховники, где он живёт, работает сеть кружков по программированию самой разной тематики. Когда Петя пошёл записываться, он увидел большой список, состоящий из  $N$  кружков. Петя хочет быть всесторонне развитой личностью, поэтому он собрался отучиться во всех этих кружках. Но когда он собрался записаться на все занятия сразу, обнаружилось, что не всё так просто. Во-первых, в один момент времени разрешается учиться только в одном из этих  $N$  кружков. Во-вторых, некоторые преподаватели выдвигают входные требования к знаниям учеников, заключающиеся в знании курсов каких-то других кружков!

Петя хочет стать великим программистом, поэтому подобные мелочи его не останавливают. Действительно, ему достаточно всего-лишь составить правильный порядок посещения кружков, чтобы удовлетворить всем входным требованиям — это совсем простая задача, доступная даже совсем неопытному программисту.

Перед тем как сесть составлять порядок посещения кружков, Петя внимательно перечитал условия обучения и обнаружил ещё один важный пункт. Оказывается, для привлечения школьников, во всех кружках действует система поощрения учеников конфетами. Это означает, что по окончании очередного кружка ученику выдают несколько коробок конфет, всё больше и больше с каждым пройденным кружком. С другой стороны, в каждом кружке количество конфет в коробке своё, зависящее от сложности курса. Более конкретно — за прохождение  $i$ -го по счёту кружка, если этот кружок идёт в общем списке под номером  $j$ , ученику выдают аж  $N^{i-1} \cdot j$  конфет — такие щедрые люди программисты.

Петя решил совместить полезное с приятным — теперь он хочет выбрать такой порядок посещения кружков, чтобы при этом получить как можно больше конфет, однако эта задача ему уже не под силу. Помогите будущему великому человеку отыскать такой порядок.

Входные данные

В первой строке входного файла содержится целое число  $N$  ( $1 \leq N \leq 100000$ ) — количество кружков в Маховниках.

В последующих  $N$  строках идут описания входных требований кружков, в порядке их следования в общем списке. В  $i$ -й строке сначала записано целое число  $k_i$  ( $0 \leq k_i \leq N - 1$ ) — количество кружков, в которых нужно отучиться перед записью в  $i$ -й кружок, а потом  $k_i$  номеров этих кружков.

Сумма  $k_i$  не превосходит 200000.

Гарантируется, что возможно посетить все эти кружки в некотором порядке, не нарушая условия посещения.

Выходные данные

Выведите  $N$  номеров, разделённых пробелами — порядок, в котором Пете надо посещать кружки, чтобы съесть как можно больше конфет.

входные данные
6 1 2 0 1 2 3 1 2 5 1 2 4 1 3 4 5
выходные данные
2 1 3 5 4 6

2A. Конденсация графа

1 секунда, 256 мегабайт

Вам задан ориентированный граф с  $N$  вершинами и  $M$  ребрами ( $1 \leq N \leq 200\,000$ ,  $1 \leq M \leq 200\,000$ ). Найдите компоненты сильной связности заданного графа и топологически отсортируйте его конденсацию.

Входные данные

4
---

Задачи - Codeforces

Граф задан во входном файле следующим образом: первая строка содержит числа  $N$  и  $M$ . Каждая из следующих  $M$  строк содержит описание ребра — два целых числа из диапазона от 1 до  $N$  — номера начала и конца ребра.

Выходные данные

На первой строке выведите число  $K$  — количество компонент сильной связности в заданном графе. На следующей строке выведите  $N$  чисел — для каждой вершины выведите номер компоненты сильной связности, которой принадлежит эта вершина. Компоненты сильной связности должны быть занумерованы таким образом, чтобы для любого ребра номер компоненты сильной связности его начала не превышал номера компоненты сильной связности его конца.

входные данные
10 19 1 4 7 8 5 10 8 9 9 6 2 6 6 2 3 8 9 2 7 2 9 7 4 5 3 6 7 3 6 7 10 8 10 1 2 9 2 7
выходные данные
2 1 2 2 1 1 2 2 2 1

2B. Расстановка чисел

1 секунда, 256 мегабайт

Вам даны  $N$  переменных  $a_1, a_2, \dots, a_N$ . Также про некоторые из этих переменных известно, что одна из них делится на другую.

Пусть мы присвоили этим переменным целые значения так, что все утверждения выполняются.

Требуется найти максимально возможное число различных значений среди этих переменных.

Входные данные

В первой строке входного файла задано два целых числа число  $N$  и  $M$  — количество переменных и количество отношений делимости ( $1 \leq N \leq 10^4$ ,  $1 \leq M \leq 10^5$ ).

Далее следуют  $M$  пар чисел  $i$  и  $j$  ( $1 \leq i, j, \leq N$ ) задающих, что  $a_i$  делится на  $a_j$ .

При этом возможны как повторяющиеся утверждения, так и тривиальные (в которых  $i = j$ ).

Выходные данные

Выведите одно число — наибольшее количество различных целых чисел в наборе  $a_i$ , удовлетворяющем всем утверждениям.

входные данные
4 6 1 1 1 2 1 3 1 4 2 3 1 3
выходные данные
4



В примере к задаче можно, например, взять  $a_1 = 0, a_2 = -4, a_3 = 2$  и  $a_4 = 3$ .

2С. Граф? А... А я думал, барон...

3 секунды, 256 мегабайт

Однажды Маша прогуливалась по парку и нашла под деревом граф... Удивлены? Вы думали, что в этой задаче будет логичная обоснованная история? Не дожидетесь! Так вот...

У Маши есть ориентированный граф, в  $i$ -й вершине которого записано некоторое целое положительное число  $a_i$ . Изначально Маша может положить в некоторую вершину графа монетку. За один ход разрешается переместить монетку, которая сейчас находится в некоторой вершине  $u$ , в любую вершину  $v$ , такую что в графе существует ориентированное ребро  $u \rightarrow v$ . Каждый раз, когда монетка оказывается в какой-либо вершине  $i$ , Маша записывает в блокнот число  $a_i$  (в частности, когда Маша изначально кладет монетку в некоторую вершину графа, она пишет в блокнот число, записанное в этой вершине). Маша хочет сделать ровно  $k - 1$  ходов таким образом, чтобы максимальное число, записанное ей в блокноте, было как можно меньше.

Входные данные

Первая строка содержит три целых числа  $n, m$  и  $k$  ( $1 \leq n \leq 2 \cdot 10^5, 0 \leq m \leq 2 \cdot 10^5, 1 \leq k \leq 10^{18}$ ) — количество вершин и ребер в графе, а также количество ходов, которое хочет сделать Маша, соответственно.

Вторая строка содержит  $n$  целых чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — числа, записанные в вершинах графа.

Каждая из следующих  $m$  строк содержит два целых числа  $u$  и  $v$  ( $1 \leq u \neq v \leq n$ ) — это означает, что в графе существует ребро  $u \rightarrow v$ .

Гарантируется, что граф не содержит петель и кратных ребер.

Выходные данные

Выведите одно целое число — минимальное значение максимального числа, которое Маша выписала в блокнот, при оптимальном перемещении монетки.

В случае, если Маше не удастся переместить монетку  $k - 1$  раз, в качестве ответа выведите число  $-1$ .

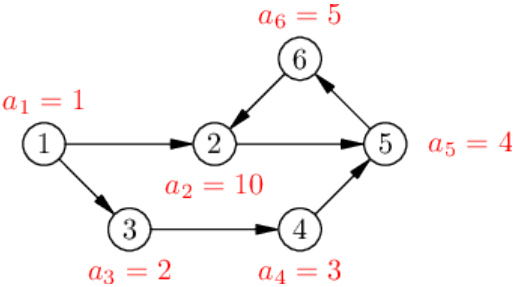
входные данные
6 7 4 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5
выходные данные
4

входные данные
6 7 100 1 10 2 3 4 5 1 2 1 3 3 4 4 5 5 6 6 2 2 5
выходные данные
10

входные данные
2 1 5 1 1 1 2
выходные данные
-1

входные данные
1 0 1 1000000000
выходные данные
1000000000

На изображении ниже приведен граф, описанный в первых двух примерах.



В первом примере можно выбрать в качестве inicialной вершину 1. После этого можно выполнить три хода:  $1 \rightarrow 3, 3 \rightarrow 4$  и  $4 \rightarrow 5$ . В итоге в блокнот будут записаны числа 1, 2, 3 и 4.

Во втором примере можно выбрать в качестве inicialной вершину 2. После этого можно выполнить 99 ходов:  $2 \rightarrow 5, 5 \rightarrow 6, 6 \rightarrow 2, 2 \rightarrow 5$ , и так далее. В итоге в блокнот будут записаны числа 10, 4, 5, 10, 4, 5, ..., 10, 4, 5, 10.

В третьем примере на имеющемся графе не удастся выполнить 4 хода.

2D. Институт

2 секунды, 512 мегабайт

Недавно Антон закончил школу, и поступил в МФТИ. Однако, он боится гулять по университетскому кампусу, потому что если он потеряет пропуск в университет, он не сможет посещать занятия.

Кампус МФТИ можно представить как ориентированный граф. По некоторым ребрам графа можно перемещаться только при наличии пропуска в университет.

Для того, чтобы потерять пропуск, Антон должен выйти из общежития, которое расположено в вершине с номером 1, затем пройти по некоторому (возможно, нулевому) количеству ребер, затем оставить пропуск в некоторой вершине, после чего продолжить гулять по кампусу. В итоге у Антона не должно быть возможности вернуться в вершину, где он оставил пропуск.

Антону интересно, не напрасны ли его опасения: помогите ему определить, возможно ли потерять пропуск в кампусе таким образом, чтобы не было возможности вернуться в вершину, где был оставлен пропуск.

Входные данные

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 300\,000$ ) — количество вершин и ребер в графе.

Каждая из следующих  $m$  строк содержит три целых числа  $u_i, v_i$  и  $t_i$  ( $1 \leq u_i, v_i \leq n, 1 \leq t_i \leq 2$ ) — описание ребра в графе. Ребро с номером  $i$  позволяет переместиться из вершины  $u_i$  в вершину  $v_i$ . Если для прохода по ребру необходим пропуск,  $t_i = 1$ . В противном случае  $t_i = 2$ .

Данный граф может содержать петли и кратные ребра.

Выходные данные

Если Антон может потерять пропуск таким образом, чтобы не было возможности вернуться в вершину, где он был оставлен, выведите «Yes». В противном случае выведите «No».

входные данные
3 4 1 2 1 2 3 2 3 2 1 3 1 2
выходные данные
Yes

входные данные
6 8 1 2 1 2 3 2 3 2 2 3 4 1 4 1 2 1 5 2 5 4 2 6 1 2
выходные данные
No

В первом примере Антон может пройти по ребру  $1 \rightarrow 2$ , воспользовавшись пропуском. Затем он может оставить пропуск в вершине 2 и пройти по ребру  $2 \rightarrow 3$ . После этого он не сможет вернуться в вершину 2.

2E. Траффик

2 секунды, 256 мегабайт

Центр Гдыни расположен на острове посередине реки Каца. Каждое утро тысячи машин едут из жилых районов на западном берегу в индустриальные на восточном.

Остров можно представить в виде прямоугольника  $A \times B$  со сторонами, параллельными осям координат, углами  $(0, 0)$  и  $(A, B)$ .

На острове есть  $n$  перекрёстков, пронумерованных натуральными числами от 1 до  $n$ . Перекрёсток номер  $i$  имеет координаты  $(x_i, y_i)$ . Если координаты какого-то перекрёстка имеют вид  $(0, y)$ , то он находится на западном берегу, если  $(A, y)$  — на восточном. Перекрёстки соединены дорогами, каждая — отрезок на плоскости, они не пересекаются (кроме концов). Дороги бывают односторонними и двусторонними. Нет никаких мостов и туннелей! Некоторые дороги могут идти по краям прямоугольника.

Поскольку плотность траффика растёт, мэр города нанял Вас (кого же ещё) проверить, достаточно ли текущей сети дорог. Он попросил написать программу, которая по карте города определит для каждого перекрёстка на западном берегу, сколько из него достижимых на восточном.

Входные данные

В первой строке входных данных дано четыре целых числа  $n, m, A$  и  $B$  ( $1 \leq n \leq 300\,000, 0 \leq m \leq 900\,000, 1 \leq A \leq 10^9, 1 \leq B \leq 10^9$ ). Это количества перекрёстков, дорог и размеры города, соответственно.

В каждой из следующих  $n$  строк есть два целых числа  $x_i$  и  $y_i$  ( $0 \leq x_i \leq A, 0 \leq y_i \leq B$ ), описывающих координаты перекрёстка номер  $i$ . Перекрёстки не совпадают.

Следующие  $m$  строк описывают дороги, каждая одну. Это описание состоит из трёх чисел:  $c_i$  — номер перекрёстка, откуда ( $1 \leq c_i \leq n$ ),  $d_i$  — куда ( $1 \leq d_i \leq n$ ),  $k_i \in \{1, 2\}$  — тип дороги (сколькосторонняя). Разные дороги соединяют разные неупорядоченные пары перекрёстков.

Задачи - Codeforces

Есть хотя бы один перекрёсток на западном берегу, из которого можно добраться хотя бы в один на восточном по дорогам.

Выходные данные

Выведите для каждого перекрёстка на западном берегу в своей строчке количество достижимых из него перекрёстков на восточном берегу.

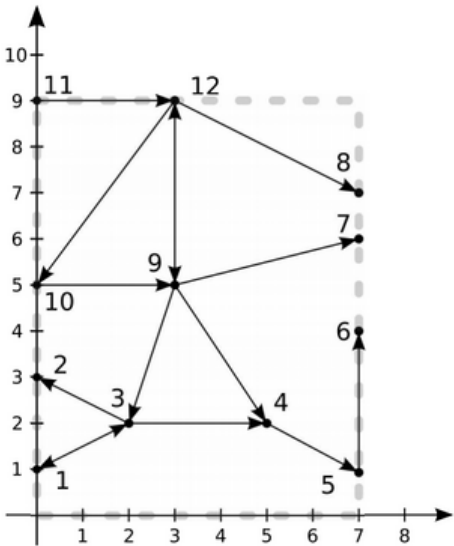
Выводите в порядке уменьшения  $y$ -координаты.

Система оценки

- 1. (30 баллов)  $n, m \leq 6\,000$ .
- 2. (70 баллов)  $n \leq 300\,000, m \leq 900\,000$ .

входные данные
5 3 1 3 0 0 0 1 0 2 1 0 1 1 1 4 1 1 5 2 3 5 2
выходные данные
2 0 2

входные данные
12 13 7 9 0 1 0 3 2 2 5 2 7 1 7 4 7 6 7 7 3 5 0 5 0 9 3 9 1 3 2 3 2 1 3 4 1 4 5 1 5 6 1 9 3 1 9 4 1 9 7 1 9 12 2 10 9 1 11 12 1 12 8 1 12 10 1
выходные данные
4 4 0 2



2F. Ориентация

2 секунды, 1024 мегабайта

Дан неориентированный граф, состоящий из  $N$  вершин и  $M$  ребер. Вершины пронумерованы целыми числами от 1 до  $N$ , а  $i$ -е ребро соединяет вершины  $a_i$  и  $b_i$ . Также дана последовательность из  $N$  целых чисел  $c_1, c_2, \dots, c_N$ .

Вы должны заменить каждое неориентированное ребро  $(a_i, b_i)$  из графа на ориентированное ребро  $a_i \rightarrow b_i$ , либо на ориентированное ребро  $b_i \rightarrow a_i$  таким образом, чтобы было выполнено следующее условие: для любого  $i$  от 1 до  $N$  существует ровно  $c_i$  вершин, достижимых из вершины  $i$ , включая саму вершину  $i$ .

Входные данные

Первая строка содержит два целых числа  $N$  и  $M$  ( $1 \leq N \leq 100$ ,  $0 \leq M \leq \frac{N(N-1)}{2}$ ).

Каждая из следующих  $M$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq N$ ). Гарантируется, что граф не содержит петель и кратных ребер.

Последняя строка содержит  $N$  целых чисел  $c_1, c_2, \dots, c_N$  ( $1 \leq c_i \leq N$ ).

Гарантируется, что для каждого теста существует решение.

Выходные данные

Выведите  $M$  строк. В  $i$ -й строке выведите «->» (без кавычек), если вместо  $i$ -го ребра нужно добавить в граф ориентированное ребро  $a_i \rightarrow b_i$ , либо «<-» (без кавычек) в противном случае.

Если существует несколько решений, выведите любое.

входные данные
3 3 1 2 2 3 3 1 3 3 3
выходные данные
-> -> ->

входные данные
3 2 1 2 2 3 1 2 3

выходные данные
<- <-

входные данные
6 3 1 2 4 3 5 6 1 2 1 2 2 1

выходные данные
<- -> ->

В первом примере получился цикл длины 3. Из каждой вершины достижима каждая вершина.

2G. 2-SAT

2 секунды, 256 мегабайт

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $(x_{i_1} = e_1) \vee (x_{i_2} = e_2)$  обратились в истину. В данной задаче вам гарантируется, что решение существует.

Входные данные

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $(x_{i_1} = e_1) \vee (x_{i_2} = e_2)$  ( $0 \leq i_j < n, 0 \leq e_j \leq 1$ ). Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

Выходные данные

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

входные данные
1 0 2 2 0 0 1 0 0 1 1 1 3 4 0 1 1 0 0 0 2 1 1 1 2 0 0 0 0 1
выходные данные
0 01 000

2H. Установка модулей GAIA

2.5 секунд, 256 мегабайт

Всего есть  $n$  модулей системы GAIA, таких как MINERVA, AETHER и другие. Модули пронумерованы от 1 до  $n$ , и для них есть ровно  $n$  слотов для подключения их к GAIA. Изначально модуль номер  $i$  подключен к слоту номер  $i$ .

Существуют только две операции, позволяющие оперировать назначением модулей GAIA по слотам. Эти операции могут быть описаны перестановками  $p$  и  $q$  длины  $n$ . В соответствии с операцией  $p$ , модуль, подключенный к слоту  $p_i$ , перемещается в слот  $i$ . Аналогично для  $q$ : при применении операции  $q$ , модуль, подключенный к слоту  $q_i$ , переподключается к слоту  $i$ .

Чтобы GAIA функционировала корректно, требуется назначить каждому модулю слот, используя *частичную композицию* операций  $p$  и  $q$ . Это означает, что для каждого  $i$  к слоту номер  $i$  должен быть подключен

- либо модуль, подключаемый к нему применением операции  $p$ ;
- либо модуль, подключаемый к нему применением операции  $p$ , а затем операции  $q$ .

Иными словами, к слоту номер  $i$  может быть подключен либо модуль с номером  $p_i$ , либо модуль с номером  $(q \circ p)_i = q_{p_i}$ . Для каждого  $i$  этот выбор можно сделать независимо от других.

Помимо этого, известны также  $m$  системных ограничений вида «модуль номер  $a_i$  не может располагаться на соседнем слоте с модулем  $b_i$ ».

Определите, существует ли частичная композиция перестановок  $p$  и  $q$ , обеспечивающая корректное функционирование GAIA, то есть при которой

- каждый модуль подключен к своему слоту, и каждый слот занят только одним модулем;
- и удовлетворены все ограничения на расположение модулей в соседних слотах.

Входные данные

В первой строке ввода через пробел даны два целых числа  $n$  и  $m$  — количество модулей системы и количество ограничений ( $1 \leq n \leq 2 \cdot 10^5; 0 \leq m \leq 2 \cdot 10^5$ ).

Во второй и третьей строках через пробел перечислены элементы перестановок  $p$  и  $q$ , описывающих операции ( $1 \leq p_i, q_i \leq n$ ). Гарантируется, что каждое число от 1 до  $n$  встречается в описании каждой операции ровно один раз.

В следующих  $m$  строках даны ограничения на расположение модулей: в  $i$ -й из них через пробел даны два целых числа  $a_i$  и  $b_i$  — номера модулей, которые не должны располагаться в соседних слотах ( $1 \leq a_i, b_i \leq n; a_i \neq b_i$ ).

Выходные данные

Если невозможно построить удовлетворяющее условию назначение, выведите «-1» (без кавычек).

Иначе выведите через пробел  $n$  целых чисел,  $i$ -е из которых равно 1, если для  $i$ -го слота выбрано назначение, соответствующее  $p$ , и 2, если выбрано назначение, соответствующее  $q \circ p$ .

Если есть несколько подходящих вариантов назначений, выведите любой из них.

Система оценки

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

Подзадача	Баллы	Дополнительные ограничения	Необходимые подзадачи	Информация о проверке
1	9	$n \leq 3$	—	полная
2	14	$n \leq 18$	1	полная
3	14	$p_i = i$ для всех $i$	—	полная
4	17	$m \leq 1$	—	полная
5	21	$q_{p_i} = i$ для всех $i$	—	полная
6	25	нет	1 – 5	первая ошибка

входные данные
3 1 3 1 2 2 1 3 1 3
выходные данные
1 2 2

входные данные
3 1 1 2 3 3 2 1 1 2
выходные данные
-1

входные данные
4 2 3 4 1 2 4 1 3 2 1 2 3 4
выходные данные
1 2 2 2

3A. Мосты

1 секунда, 256 мегабайт

Дан неориентированный граф. Требуется найти все мосты в нём.

Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000, 1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Рёбро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

Выходные данные

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество мостов в заданном графе.

На следующей строке выведите  $b$  целых чисел — номера рёбер, которые являются мостами, в возрастающем порядке. Рёбра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

входные данные
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6
выходные данные
1 3

3B. Точки сочленения

2 секунды, 256 мегабайт

Дан неориентированный граф. Требуется найти все точки сочленения в нём.

Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 20\,000, 1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Рёбро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

Выходные данные

Первая строка выходного файла должна содержать одно натуральное число  $b$  — количество точек сочленения в заданном графе. На следующей строке выведите  $b$  целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

входные данные
6 7 1 2 2 3 2 4 2 5 4 5 1 3 3 6
выходные данные
2 2 3

3C. Магнитные подушки

2 секунды, 256 мегабайт

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добираться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

Входные данные

В первой строке входного файла находятся числа  $N$  и  $M$  — количество небоскребов в городе и количество работающих магнитных подушек соответственно ( $3 \leq N \leq 100\,000, 1 \leq M \leq 100\,000$ ). В каждой из следующих  $M$  строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до  $N$ . Гарантируется, что имеющиеся магнитные подушки позволяют перемещаться от одного небоскреба до любого другого.

Выходные данные

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

входные данные
3 1 1 2 3
выходные данные
1 1

входные данные
3 2 1 2 3 3 2 1
выходные данные
0

входные данные
5 4 1 2 3 2 4 3 1 2 4 3 5 1
выходные данные
1 4

3D. Размещение данных

1 секунда, 256 мегабайт

Телекоммуникационная сеть крупной IT-компании содержит  $n$  серверов, пронумерованных от 1 до  $n$ . Некоторые пары серверов соединены двусторонними каналами связи, всего в сети  $m$  каналов. Гарантируется, что сеть серверов устроена таким образом, что по каналам связи можно передавать данные с любого сервера на любой другой сервер, возможно с использованием одного или нескольких промежуточных серверов.

Множество серверов  $A$  называется отказоустойчивым, если при недоступности любого канала связи выполнено следующее условие. Для любого не входящего в это множество сервера  $X$  существует способ передать данные по остальным каналам на сервер  $X$  хотя бы от одного сервера из множества  $A$ .

На рис. 1 показан пример сети и отказоустойчивого множества из серверов с номерами 1 и 4. Данные на сервер 2 можно передать следующим образом. При недоступности канала между серверами 1 и 2 — с сервера 4, при недоступности канала между серверами 2 и 3 — с сервера 1. На серверы 3 и 5 при недоступности любого канала связи можно по другим каналам передать данные с сервера 4.

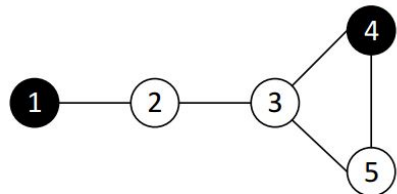


Рис. 1. Пример сети и отказоустойчивого множества серверов.

В рамках проекта группе разработчиков компании необходимо разместить свои данные в сети. Для повышения доступности данных и устойчивости к авариям разработчики хотят продублировать свои данные, разместив их одновременно на нескольких серверах, образующих отказоустойчивое множество. Чтобы минимизировать издержки, необходимо выбрать минимальное по количеству серверов отказоустойчивое множество. Кроме того, чтобы узнать, насколько гибко устроена сеть, необходимо подсчитать количество способов выбора такого множества, и поскольку это количество способов может быть большим, необходимо найти остаток от деления этого количества способов на число  $10^9 + 7$ .

Требуется написать программу, которая по заданному описанию сети определяет следующие числа:  $k$  — минимальное количество серверов в отказоустойчивом множестве серверов,  $c$  — остаток от деления количества способов выбора отказоустойчивого множества из  $k$  серверов на число  $10^9 + 7$

**Входные данные**

Первая строка входного файла содержит целые числа  $n$  и  $m$  — количество серверов и количество каналов связи соответственно ( $2 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$ ). Следующие  $m$  строк содержат по два целых числа и описывают каналы связи между серверами. Каждый канал связи задается двумя целыми числами: номерами серверов, которые он соединяет.

Гарантируется, что любые два сервера соединены напрямую не более чем одним каналом связи, никакой канал не соединяет сервер сам с собой, и для любой пары серверов существует способ передачи данных с одного из них на другой, возможно с использованием одного или нескольких промежуточных серверов.

**Выходные данные**

Выведите два целых числа, разделенных пробелом:  $k$  — минимальное число серверов в отказоустойчивом множестве серверов,  $c$  — количество способов выбора отказоустойчивого множества из  $k$  серверов, взятое по модулю  $10^9 + 7$ .

входные данные
5 5 1 2 2 3 3 4 3 5 4 5
выходные данные
2 3

3Е. Компоненты реберной двусвязности

2 секунды, 64 мегабайта

Компонентой реберной двусвязности графа  $\langle V, E \rangle$  называется подмножество вершин  $S \subset V$ , такое что для любых различных  $u$  и  $v$  из этого множества существует не менее двух реберно не пересекающихся путей из  $u$  в  $v$ .

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

**Входные данные**

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

**Выходные данные**

В первой строке выходного файла выведите целое число  $k$  — количество компонент реберной двусвязности графа.

Во второй строке выведите  $n$  натуральных чисел  $a_1, a_2, \dots, a_n$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты реберной двусвязности, которой принадлежит  $i$ -я вершина.

Компоненты требуется нумеровать в порядке возрастания минимального номера вершины, входящей в компоненту.

входные данные
6 7 1 2 2 3 3 1 1 4 4 5 4 6 5 6
выходные данные
2 1 1 1 2 2 2

3F. Компоненты вершинной двусвязности

2 секунды, 64 мегабайта

Компонентой вершинной двусвязности графа  $\langle V, E \rangle$  называется максимальный по включению подграф (состоящий из вершин и ребер), такой что любые два ребра из него лежат на вершинно простом цикле.

Дан неориентированный граф без петель. Требуется выделить компоненты вершинной двусвязности в нем.

**Входные данные**

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и ребер графа соответственно ( $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 200\,000$ ).

Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается двумя натуральными числами  $b_i, e_i$  — номерами концов ребра ( $1 \leq b_i, e_i \leq n$ ).

**Выходные данные**

В первой строке выходного файла выведите целое число  $k$  — количество компонент вершинной двусвязности графа.

Во второй строке выведите  $m$  натуральных чисел  $a_1, a_2, \dots, a_m$ , не превосходящих  $k$ , где  $a_i$  — номер компоненты вершинной двусвязности, которой принадлежит  $i$ -е ребро. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Компоненты требуется нумеровать в порядке возрастания минимального номера ребра, входящего в компоненту.

входные данные
5 6 1 2 2 3 3 1 1 4 4 5 5 1
выходные данные
2 1 1 1 2 2 2

4А. Разбиение на пути

1 секунда, 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество реберно-простых путей.

**Входные данные**

Дан граф. На первой строке число вершин  $n$  ( $1 \leq n \leq 20\,000$ ) и число рёбер  $m$  ( $1 \leq m \leq 20\,000$ ). Следующие  $m$  строк содержат описание рёбер графа. Каждая строка по два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ). Между каждыми двумя вершинами не более одного ребра. Граф связан.

**Выходные данные**

На первой строке минимальное число путей. На каждой следующей строке описание очередного пути — номера вершин в порядке прохождения.



входные данные
7 7 1 2 4 1 6 7 5 7 7 4 2 3 4 2
выходные данные
3 2 4 7 6 7 5 4 1 2 3

4В. Таня и пароль

2 секунды, 256 мегабайт

Пока папа был на работе, маленькая девочка Таня решила поиграть с папиным паролем к секретной базе данных. Папин пароль представляет собой строку, состоящую из  $n + 2$  символов. Она выписала все возможные  $n$  трёхбуквенных подстрок пароля на бумажки, по одной на каждую бумажку, а сам пароль выкинула. Каждая трёхбуквенная подстрока была выписана на бумажки столько раз, сколько она встречалась в пароле. Таким образом, в итоге у Тани оказалось  $n$  бумажек.

Потом Таня поняла, что папа расстроится, если узнает о ее игре, и решила восстановить пароль или, по крайней мере, хотя бы какую-то строку, соответствующую получившемуся набору трёхбуквенных строк. Вам предстоит помочь ей в этой непростой задаче. Известно, что папин пароль состоял из строчных и заглавных букв латинского алфавита, а также из цифр. Строчные и заглавные буквы латинского алфавита считаются различными.

Входные данные

В первой строке следует целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ), количество трёхбуквенных подстрок, которые получились у Тани.

Следующие  $n$  строк каждая содержат по три буквы, образующие подстроку пароля папы. Каждый символ во вводе — строчная или заглавная буква латинского алфавита или цифра.

Выходные данные

Если во время игры Таня что-то напутала, и строк, соответствующих данному набору подстрок, не существует, то выведите «NO».

Если же возможно восстановить строку, соответствующую данному набору подстрок, то выведите «YES», а затем любой подходящий вариант пароля.

входные данные
5 aca aba aba cab bac
выходные данные
YES abacaba

входные данные
4 abc bCb cb1 b13
выходные данные
NO

входные данные
7 aaa aaa aaa aaa aaa aaa aaa
выходные данные
YES aaaaaaaa

4С. Из истории банка Гринготтс

1 секунда, 256 мегабайт

Гоблинский банк "Гринготтс" по праву считается одним из самых надёжных банков в мире волшебников. Он находится в сотнях километров под Лондоном, и охраняется мощными заклинаниями и драконами. А знаете ли вы, как был построен этот банк? Гоблины ведь не такие дураки, как гномы — они не станут копать многокилометровые подземные ходы киркой и лопатой. Как они проложили все туннели без лопат и кирок? Очень просто: чтобы прорыть туннель, гоблины с помощью магии вызывали гипервяка. Гипервяк — это нечто вроде большущей змеи мощностью более тысячи мегачервей. Двигаясь, он поедает всё, что находится на его пути, оставляя за собой удобный, ровный и прочный коридор. Гипервяком легко управлять с помощью магии, заставляя его рыть туннели в нужном направлении.

А знаете ли вы, почему гномы и другие подземные создания не используют гипервяков? Потому, что гипервяки вообще не могут двигаться, не поедая землю или камни. Значит, если гипервяк докопал туннель до нужного вам места, и новые туннели из этого места не нужны, то единственный способ остановить гипервяка — это, ... хм ..., скажем так, дематериализовать бедную зверушку. Что конечно, строжайше запрещено, ведь эти создания внесены в оранжевую книгу редких магических существ, и за каждого положено платить большой штраф. Просто под землей, где никто не видит, гоблины не очень заботятся о законности. Зато они заботятся о надежности банка, поэтому никогда не копают больше одного туннеля между двумя комнатами. И уж конечно, гоблинам и в голову не придет копать туннель, не соединяющий две разные комнаты.

К сожалению, министру магии попала в руки полная схема туннелей между помещениями банка. Он хочет потребовать штраф за каждый прорытый туннель. Гоблины, конечно, могут заявить, что если два туннеля приходят в одну комнату, то они обошлись всего одним гипервяком для прокладывания обоих туннелей, проведя гипервяка по краю комнаты. И вообще, для любой цепочки туннелей, где конечная комната очередного туннеля совпадает с начальной комнатой следующего, хватит одного гипервяка. Но, разумеется, никто не поверит, что гоблины перетаскивали гипервяка из одного помещения в другое, а расширять существующие туннели они не могли из соображений безопасности. Помогите гоблинам представить министру магии такой план рытья туннелей, при котором погибло бы минимальное количество гипервяков.

Входные данные

В первой строке входа находятся числа  $N$  и  $M$  ( $2 \leq N \leq 20\,000; 1 \leq M \leq 20\,000$ ) — количество помещений банка и туннелей между помещениями. В следующих  $M$  строках указаны пары чисел (каждое число от 1 до  $N$ ) — номера помещений, которые соединяет очередной туннель. От каждого помещения банка можно добраться до другого, пользуясь только прогрызенными гипервяками туннелями.

Выходные данные

В первой строке вывести минимальное количество гипервяков  $K$ , необходимое для прокладывания всех туннелей. В последующих  $K$  строках через пробел последовательно перечислите номера вершин, через которые прогрызал ходы соответствующий гипервяк согласно вашей схеме.

входные данные

7 7  
1 2  
4 1  
6 7  
5 7  
7 4  
2 3  
4 2

выходные данные

3  
2 4 7 6  
7 5  
4 1 2 3

4D. Почтальон

1 секунда, 256 мегабайт

В городе есть  $N$  площадей, соединенных улицами. При этом количество улиц не превышает 100 000 и существует не более трех площадей, на которые выходит нечетное количество улиц. Для каждой улицы известна ее длина. По каждой улице разрешено движение в обе стороны. В городе есть хотя бы одна улица. От каждой площади до любой другой можно дойти по улицам.

Почтальону требуется пройти хотя бы один раз по каждой улице. Почтальон хочет, чтобы длина его пути была минимальна. Он может начать движение на любой площади и закончить также на любой (в том числе и на начальной).

Помогите почтальону составить такой маршрут.

Входные данные

Сначала записано число  $N$  — количество площадей в городе ( $2 \leq N \leq 1\,000$ ).

Далее следуют  $N$  строк, задающих улицы. В  $i$ -й из этих строк находится число  $m_i$  — количество улиц, выходящих из площади  $i$ . Далее следуют  $m_i$  пар натуральных чисел: в  $j$ -й паре первое число — номер площади, в которую идет  $j$ -я улица с  $i$ -й площади, а второе число — длина этой улицы.

Между двумя площадями может быть несколько улиц, но не может быть улицы с площади на нее саму.

Все числа во входном файле не превосходят 100 000.

Выходные данные

Если решение существует, то в первую строку выходного файла выведите одно число — количество улиц в искомом маршруте, а во вторую — номера площадей в порядке их посещения.

Если решения нет, выведите в выходной файл одно число —1.

Если решений несколько, выведите любое.

входные данные

4  
2 2 1 2 2  
4 1 2 4 4 3 5 1 1  
2 2 5 4 8  
2 3 8 2 4

выходные данные

5  
1 2 4 3 2 1

Задачи - Codeforces

Дан ориентированный невзвешенный граф. Также даны стартовая и конечная вершина. Найдите и выведите кратчайший путь между этими двум вершинами.

Входные данные

В первой строке даны  $n$  и  $m$  — число вершин и ребер в графе ( $1 \leq n \leq 10^5, 0 \leq m \leq 2 \cdot 10^5$ ). В следующих  $m$  строках идут пары чисел, задающие ребра.

В последней строке даны  $s$  и  $t$  — номера стартовой и конечной вершины ( $1 \leq s, t \leq n, s \neq t$ ).

Выходные данные

Если пути не существует, выведите «-1» без кавычек. Иначе в первой строке выведите длину кратчайшего пути, а в следующей — вершины пути, включая стартовую и конечную.

входные данные

5 5  
1 2  
2 3  
3 4  
1 5  
5 4  
1 4

выходные данные

2  
1 5 4

5B. План эвакуации

1 секунда, 256 мегабайт

Схема этажа представляет собой клетчатую сетку. Каждая клетка которой может быть либо пустой, либо занята стеной. Необходимо написать программу, которая по схеме этажа составит план эвакуации. То есть для каждой пустой клетки определит, в каком направлении от нее лучше всего двигаться, чтобы выйти за пределы этажа за наименьшее число перемещений. Перемещаться можно только между клетками, имеющими общую сторону.

Гарантируется, что из каждой пустой клетки можно выбраться за пределы этажа.

Входные данные

Первая строка содержит два целых числа  $n$  и  $m$  — количество столбцов и строк в схеме этажа соответственно ( $3 \leq n, m \leq 1\,000$ ). Следующие  $n$  строк содержат по  $m$  символов «.» и «#», обозначающих пустую клетку и клетку со стеной соответственно.

Выходные данные

Ваша программа должна вывести  $n$  строк по  $m$  символов — оригинальный план этажа, в котором каждая пустая клетка заменена на символ, обозначающий одно из четырех направлений: движение вниз («v», ASCII 118), вверх («^», ASCII 94), влево («<», ASCII 60), вправо («>», ASCII 62).

Если ответов на задачу несколько, можете вывести любой из них.

входные данные

5 5  
.....  
.....  
.....  
.....  
.....

выходные данные

<^^^  
<^^>  
<<v>>  
<<vv>  
vvvv>

5A. Кратчайший путь в невзвешенном графе

1 секунда, 256 мегабайт

входные данные
5 6 ..#### ..#..# ###.## .#.... .#....
выходные данные
<^#### <<#v<# ###v## <#vvv> v#vvv>

5C. Правила дорожного движения

1.5 секунд, 256 мегабайт

В столице одной небольшой страны очень сложная ситуация. Многокилометровые пробки буквально парализовали движение в городе, и власти на многих улицах ввели одностороннее движение, не анализируя, можно ли будет теперь проехать из любого места в городе в любое другое, не нарушая правила. Транспортная система столицы представляет собой  $N$  площадей, соединенных  $M$  полосами для движения, в том числе круговыми полосами, проходящими по площади. Каждая полоса предназначена для движения только в одну определенную сторону. При этом на магистралях есть полосы, направленные как в одну, так и в другую сторону. По круговой полосе можно двигаться только внутри площади и только против часовой стрелки.

Власти города на каждой полосе разместили видеокамеру, поэтому если Иннокентий едет по встречной полосе (при ее наличии) или, в случае одностороннего движения, в сторону противоположную предписанной знаками, то после поездки против правил по каждой из полос ему придется заплатить штраф в размере одной тысячи тугриков этой страны.

Иннокентий, который торопится купить кафельную плитку со скидкой, решил съездить до магазина в любом случае, даже если для этого придется нарушать правила. Но он хочет выбрать такой маршрут движения, суммарный штраф на котором минимален.

Иннокентий еще не решил, откуда именно и в какой магазин он собирается ехать, поэтому ему необходимо ответить на несколько вопросов вида «Какой минимальный штраф надо заплатить, чтобы добраться из пункта  $A$  в пункт  $B$ ?». Отвечая на потребности жителей столицы, известная поисковая система Индекс разрабатывает соответствующий сервис.

Так как многие из вас рано или поздно будут проходить собеседование на работу в эту фирму, продемонстрируйте, что вы тоже умеете решать эту задачу.

**Входные данные**  
В первой строке входных данных содержатся два числа  $N$  и  $M$  — количество площадей и полос движения в городе соответственно ( $1 \leq N \leq 5000, 1 \leq M \leq 10\,000$ ). Далее содержатся описания полос, по которым движение разрешено. Каждая полоса описывается номерами двух площадей, которые она соединяет. Движение разрешено в направлении от первой из указанных площадей ко второй.

В следующей строке содержится одно число  $K$  — количество вопросов у Иннокентия ( $1 \leq K \leq 10\,000, N \cdot K \leq 2 \cdot 10^7$ ). В следующих строках описываются вопросы, каждый вопрос описывается номерами двух площадей, между которыми требуется найти самый дешевый путь. Путь необходимо проложить от первой из указанных площадей ко второй.

**Выходные данные**  
Для каждого вопроса выведите одно число — искомый минимальный размер штрафа в тысячах тугриков. В случае, если пути между выбранной парой площадей не существует, выведите  $-1$ .

входные данные
5 5 2 1 2 4 3 2 4 3 5 4 3 5 1 1 5 2 3
выходные данные
0 2 0

5D. Числа

1 секунда, 256 мегабайт

Витя хочет придумать новую игру с числами. В этой игре от игроков требуется преобразовывать четырехзначные числа не содержащие нулей при помощи следующего разрешенного набора действий:

- 1. Можно увеличить первую цифру числа на 1, если она не равна 9.
- 2. Можно уменьшить последнюю цифру на 1, если она не равна 1.
- 3. Можно циклически сдвинуть все цифры на одну вправо.
- 4. Можно циклически сдвинуть все цифры на одну влево.

Например, применяя эти правила к числу 1234, можно получить числа 2234, 1233, 4123 и 2341 соответственно.

Точные правила игры Витя пока не придумал, но пока его интересует вопрос, как получить из одного числа другое за минимальное количество операций.

**Входные данные**  
Во входном файле содержится два различных четырехзначных числа, каждое из которых не содержит нулей.

**Выходные данные**  
Программа должна вывести последовательность четырехзначных чисел, не содержащих нулей. Последовательность должна начинаться первым из данных чисел и заканчиваться вторым из данных чисел, каждое последующее число в последовательности должно быть получено из предыдущего числа применением одного из правил. Количество чисел в последовательности должно быть минимально возможным.

входные данные
9876 8876
выходные данные
9876 8769 8768 8876

входные данные
1234 4321
выходные данные
1234 2234 3234 4323 4322 4321

5E. Найти короткий путь

1 секунда, 256 мегабайт

входные данные
3 0 1 0 0 1 3 2 3
выходные данные
6

Однажды Юрий решил снарядить очередной поход в горы и собрался взять с собой несколько друзей. Горы можно представить как  $n$  полян, между которыми есть тропы. Юрий знает тропы двух типов — короткие и длинные. По каждой тропе можно ходить только в одном направлении. Правда, длинные тропы не всегда длиннее коротких.

Для того, чтобы друзья не считали своего проводника скучным человеком, Юрий всегда чередует тропы. Строго говоря, Юрий никогда не ходит по двум тропам одного типа подряд. Юрий рассматривает несколько маршрутов через горы, каждый из которых заканчивается на поляне с номером  $n$ , от которой можно добраться до города. Не смотря на длину троп, каждую из них Юрий проходит за одну минуту. Для каждого варианта начальной поляны посчитайте минимальное время прогулки от текущей поляны до поляны с номером  $n$ , если Юрий верен своим принципам и никогда не ходит по двум тропам одного типа подряд.

**Входные данные**  
В первой строке через пробел заданы два числа  $n$  и  $m$  — количество полян и троп соответственно ( $2 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$ ).  
В следующих  $m$  строках записаны три числа:  $a_i, b_i$  и  $type_i$ , означающие, что с поляны  $a_i$  на поляну  $b_i$  ведет тропа типа  $type_i$ , где  $type_i \in \{1, 2\}$  ( $type_i$  означает тип тропы).  
Обратите внимание, что тропа может соединять поляну саму с собой, а также между двумя полянами может быть несколько троп.

**Выходные данные**  
Выведите через пробел  $n - 1$  число — минимальное время прогулки от вершины с номером  $i$  до вершины с номером  $n$ . Если от какой-то вершины нельзя добраться до вершины с номером  $n$ , выполняя принципы Юрия, выведите вместо минимального времени число  $-1$ .

входные данные
6 6 1 2 1 2 3 2 3 1 1 1 6 2 4 3 1 5 1 2
выходные данные
1 3 2 -1 5

6A. Пифагоров экспресс (для C++)

2.5 секунд, 256 мегабайт

В стране Флатландии есть  $n$  городов, расположенных в целочисленных точках плоскости. Транспортная система Флатландии настолько развита, что между любыми двумя городами ходит экспресс имени Пифагора. С помощью него можно добраться от города с координатами  $x_1, y_1$  до города с координатами  $x_2, y_2$  за время  $(x_1 - x_2)^2 + (y_1 - y_2)^2$ . По каждой линии ходит достаточно поездов, и временем на пересадки можно пренебречь.  
Сообщение с внешним миром во Флатландии продумано несколько хуже, и единственный аэропорт международного сообщения находится в городе с номером  $s$ . Вам же хочется попасть в город с номером  $t$ . Определите, за какое наименьшее время это можно сделать.

**Входные данные**  
В первой строке задано число городов  $n$  ( $1 \leq n \leq 2 \cdot 10^4$ ). В следующих  $n$  строках заданы координаты городов  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^4$ ). В последней строке даны  $s$  и  $t$  — номера начального и конечного города в пути ( $1 \leq s, t \leq n$ ).  
**Выходные данные**  
Выведите одно число — минимальное время, за которое можно добраться из  $s$  в  $t$ .

6B. Флойд

1 секунда, 256 мегабайт

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать матрицу длин кратчайших путей от каждой вершины до всех остальных вершин.

**Входные данные**  
Программа получает сначала число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа и число  $M$  ( $0 \leq M \leq 10^4$ ) — количество ребер.  
В следующих строках идет  $M$  троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от  $-100$  до  $100$ ).

**Выходные данные**  
Требуется вывести матрицу  $N$  на  $N$ . В  $j$ -м столбце  $i$ -й строки следует вывести длину кратчайшего пути от вершины  $i$  до вершины  $j$ , либо число  $30\,000$ , если не существует пути между данными вершинами.

входные данные
6 4 1 2 10 2 3 10 1 3 100 4 5 -10
выходные данные
0 10 20 30000 30000 30000 30000 0 10 30000 30000 30000 30000 30000 0 30000 30000 30000 30000 30000 30000 0 -10 30000 30000 30000 30000 30000 0 30000 30000 30000 30000 30000 30000 0

6C. Форд-Беллман

1 секунда, 256 мегабайт

Дан ориентированный граф, в котором могут быть кратные ребра и петли. Каждое ребро имеет вес, выражающийся целым числом (возможно, отрицательным). Гарантируется, что циклы отрицательного веса отсутствуют.

Требуется посчитать длины кратчайших путей от вершины номер 1 до всех остальных вершин.

**Входные данные**  
Программа получает сначала число  $N$  ( $1 \leq N \leq 2\,000$ ) — количество вершин графа и число  $M$  ( $0 \leq M \leq 10^4$ ) — количество ребер.  
В следующих строках идет  $M$  троек чисел, описывающих ребра: начало ребра, конец ребра и вес (вес — целое число от  $-100$  до  $100$ ).  
**Выходные данные**  
Программа должна вывести  $N$  чисел — расстояния от вершины номер 1 до всех вершин графа. Если пути до соответствующей вершины не существует, вместо длины пути выведите число  $30\,000$ .

входные данные
6 4 1 2 10 2 3 10 1 3 100 4 5 -10
выходные данные
0 10 20 30000 30000 30000

6D. Кратчайший путь – 2

2 секунды, 256 мегабайт

Дан неориентированный связный взвешенный граф. Найдите кратчайшее расстояние от первой вершины до всех вершин.

Входные данные

В первой строке входного файла два числа:  $n$  и  $m$  ( $2 \leq n \leq 30000, 1 \leq m \leq 400000$ ), где  $n$  — количество вершин графа, а  $m$  — количество ребер.

Следующие  $m$  строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — неотрицательное целое число, не превосходящее  $10^4$ .

Выходные данные

Выведите  $n$  чисел — для каждой вершины кратчайшее расстояние до нее.

входные данные
4 5 1 2 1 1 3 5 2 4 8 3 4 1 2 3 3
выходные данные
0 1 4 5

6E. Транзитивное замыкание

1 секунда, 256 мегабайт

Невзвешенный ориентированный граф задан своей матрицей смежности.

Требуется построить его транзитивное замыкание, то есть матрицу, в которой в  $i$ -й строке и  $j$ -м столбце находится 1, если от вершины  $i$  можно добраться до вершины  $j$ , и 0 — иначе.

Входные данные

В первой строке дано число  $N$  ( $1 \leq N \leq 100$ ) — число вершин в графе.

Далее задана матрица смежности графа: в  $N$  строках даны по  $N$  чисел 0 или 1 в каждой:  $i$ -е число в  $i$ -й строке всегда равно 1.

Выходные данные

Необходимо вывести матрицу транзитивного замыкания графа в формате, аналогичным формату матрицы смежности.

входные данные
4 1 1 0 0 0 1 1 0 1 0 1 0 0 0 1 1
выходные данные
1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 1

6F. Есть ли отрицательный цикл

1 секунда, 256 мегабайт

Дан ориентированный граф. Определить, есть ли в нем цикл отрицательного веса.

Входные данные

В первой строке содержится число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Веса ребер по модулю меньше  $10^5$ . Если ребра нет, соответствующее значение равно  $10^5$ .

Выходные данные

Выведите «YES», если цикл существует, или «NO», в противном случае.

входные данные
3 100000 100000 -51 100 100000 100000 100000 -50 100000
выходные данные
YES

входные данные
3 100000 100000 -49 100 100000 100000 100000 -50 100000
выходные данные
NO

6G. Кратчайший путь длины K

4 секунды, 256 мегабайт

Дан ориентированный граф. Найдите кратчайшие пути, состоящие из  $K$  ребер, от  $S$  до всех вершин.

Входные данные

В первой строке дано целых четыре целых числа:  
 $1 \leq N, M \leq 10^4$  — количества вершин и ребер,  $0 \leq K \leq 100$  — количество ребер в кратчайших путях,  $1 \leq S \leq N$  — начальная вершина.

В последующих  $M$  строках даны тройки целых чисел  $a_i, b_i, w$  — начало и конец ребра, а также его вес ( $1 \leq a_i, b_i \leq N, -10^5 \leq w \leq 10^5$ ).

Выходные данные

Выведите ровно  $N$  чисел по одному в строке.  $i$ -е число — длина минимального пути из ровно  $K$  ребер из  $S$  в  $i$ , или  $-1$ , если пути не существует.

входные данные
3 3 1 1 1 2 100 2 3 300 1 3 2
выходные данные
-1 100 2

входные данные
3 3 2 1 1 2 100 2 3 300 1 3 2

выходные данные
-1 -1 400

6H. Цикл отрицательного веса

2 секунды, 256 мегабайт

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

Входные данные

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

Выходные данные

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

входные данные
2 0 -1 -1 0
выходные данные
YES 2 2 1

6I. Егор и граф

3 секунды, 256 мегабайт

У Егора есть взвешенный ориентированный граф, состоящий из  $n$  вершин. В этом графе между любой парой различных вершин есть ребро в обоих направлениях. Егор любит играть с графом, и сейчас он придумал новую игру:

- Игра состоит из  $n$  шагов.
- На  $i$ -том шаге Егор удаляет из графа вершину номер  $x_i$ . Удаляя вершину, Егор удаляет все ребра, которые входили в данную вершину и которые выходили из нее.
- Перед выполнением каждого шага, Егор хочет знать сумму длин кратчайших путей между всеми парами оставшихся вершин. Кратчайший путь может проходить через любую оставшуюся вершину. Другими словами, если обозначить как  $d(i, v, u)$  кратчайший путь между вершинами  $v$  и  $u$  в графе, который получился до удаления вершины  $x_i$ , то Егор хочет знать значение следующей суммы:  $\sum_{v, u, v \neq u} d(i, v, u)$ .

Помогите Егору, выведите значение искомой суммы перед каждым шагом.

Входные данные

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 500$ ) — количество вершин в графе.

В следующих  $n$  строках содержится по  $n$  целых чисел — матрица смежности графа:  $j$ -тое число в  $i$ -той строке  $a_{ij}$  ( $1 \leq a_{ij} \leq 10^5, a_{ii} = 0$ ) обозначает вес ребра, ведущего из вершины  $i$  в вершину  $j$ .

В следующей строке содержится  $n$  различных целых чисел:  $x_1, x_2, \dots, x_n$  ( $1 \leq x_i \leq n$ ) — вершины, которые удаляет Егор.

Выходные данные

Задачи - Codeforces

Выведите  $n$  целых чисел —  $i$ -тое число равно искомой сумме перед  $i$ -тым шагом.

Пожалуйста, не используйте спецификатор `%lld` для чтения или записи 64-х битовых чисел на C++. Рекомендуется использовать потоки `cin`, `cout` или спецификатор `%I64d`.

входные данные
1 0 1
выходные данные
0

входные данные
2 0 5 4 0 1 2
выходные данные
9 0

входные данные
4 0 3 1 1 6 0 400 1 2 4 0 1 1 1 1 0 4 1 2 3
выходные данные
17 23 404 0

7A. Остовное дерево

8 секунд, 256 мегабайт

Даны точки на плоскости, являющиеся вершинами полного графа. Вес ребра равен расстоянию между точками, соответствующими концам этого ребра. Требуется в этом графе найти остовное дерево минимального веса.

Входные данные

Первая строка входного файла содержит натуральное число  $n$  — количество вершин графа ( $1 \leq n \leq 10\,000$ ). Каждая из следующих  $n$  строк содержит два целых числа  $x_i, y_i$  — координаты  $i$ -й вершины ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие две точки не совпадают.

Выходные данные

Первая строка выходного файла должна содержать одно вещественное число — вес минимального остовного дерева.

входные данные
2 0 0 1 1
выходные данные
1.4142135624

7B. Остовное дерево 2

2 секунды, 256 мегабайт

Требуется найти в связном графе остовное дерево минимального веса.

Входные данные



Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100\,000$ ).  $n \leq 200\,000$ ,  $m \leq 200\,000$ .

Граф является связным.

Выходные данные

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

входные данные
4 4 1 2 1 2 3 2 3 4 5 4 1 4
выходные данные
7

7C. Уничтожение массива

1 секунда, 256 мегабайт

Вам дан массив, состоящий из  $n$  неотрицательных целых чисел  $a_1, a_2, \dots, a_n$ .

В этом массиве один за другим зачёркиваются числа. Вам задана перестановка чисел от 1 до  $n$  — порядок, в котором это происходит.

После зачёркивания очередного числа вам необходимо найти в этом массиве подотрезок с максимальной суммой, не содержащий ни одного зачёркнутого числа. Сумму чисел в пустом подотрезке считайте равной 0.

Входные данные

В первой строке входных данных записано число  $n$  ( $1 \leq n \leq 100\,000$ ) — длина массива.

В второй строке записаны  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ ).

В третьей строке входных данных записана перестановка чисел от 1 до  $n$  — порядок, в котором зачеркиваются числа.

Выходные данные

В выходной файл выведите  $n$  строк, каждая из которых должна содержать одно число — максимальную сумму на подотрезке заданного массива, не содержащем зачёркнутых чисел, после выполнения очередного действия.

входные данные
4 1 3 2 5 3 4 1 2
выходные данные
5 4 3 0

входные данные
5 1 2 3 4 5 4 2 3 5 1
выходные данные
6 5 5 1 0

входные данные
8 5 5 4 4 6 6 5 5 5 2 8 7 1 3 4 6
выходные данные
18 16 11 8 8 6 6 0

В первом тестовом примере происходит следующее:

- 1. Зачеркивается третий элемент, массив принимает вид 1 3 \* 5. Отрезок с максимально суммой 5 состоит из одного числа 5.
- 2. Зачеркивается четвертый элемент, массив принимает вид 1 3 \* \*. Отрезок с максимально суммой 4 состоит из двух чисел 1 3.
- 3. Зачеркивается первый элемент, массив принимает вид \* 3 \* \*. Отрезок с максимально суммой 3 состоит из одного числа 3.
- 4. Зачеркивается оставшийся второй элемент, в этот момент непустых допустимых подотрезков не остается, поэтому здесь ответ равен нулю.

7D. Разрезание графа

2 секунды, 256 мегабайт

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- cut — разрезать граф, то есть удалить из него ребро;
- ask — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа cut рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа ask.

Входные данные

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа  $n$ , количество рёбер  $m$  и количество операций  $k$  ( $1 \leq n \leq 50\,000$ ,  $0 \leq m \leq 100\,000$ ,  $m \leq k \leq 150\,000$ ).

Следующие  $m$  строк задают рёбра графа;  $i$ -я из этих строк содержит два числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), разделённые пробелами — номера концов  $i$ -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют  $k$  строк, описывающих операции. Операция типа cut задаётся строкой "cut  $u$   $v$ " ( $1 \leq u, v \leq n$ ), которая означает, что из графа удаляют ребро между вершинами  $u$  и  $v$ . Операция типа ask задаётся строкой "ask  $u$   $v$ " ( $1 \leq u, v \leq n$ ), которая означает, что необходимо узнать, лежат ли в данный момент вершины  $u$  и  $v$  в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа cut ровно один раз.

Выходные данные

Для каждой операции ask во входном файле выведите на отдельной строке слово "YES", если две указанные вершины лежат в одной компоненте связности, и "NO" в противном случае. Порядок ответов должен соответствовать порядку операций ask во входном файле.

1 секунда, 256 мегабайт

Ценное дерево с  $n$  вершинами растет возле дома Байтазара. Ребро  $i$  имеет цену  $v_i$ .

Определим поддерево дерева как непустое **связное** подмножество ребер.

*Цена* поддерева определяется, как количество ребер в поддереве, умноженное на цену **минимального** ребра поддерева.

Байтазар хочет зарабатывать деньги на продаже поддереьев, поэтому он хочет знать максимальную цену поддерева его дерева.

Входные данные

В первой строке записано число  $n$  ( $2 \leq n \leq 10^5$ ) — количество вершин в дереве.

В каждой из следующих  $n - 1$  строк записаны три числа  $a_i, b_i$  и  $v_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i, 1 \leq v_i \leq 10^9$ ) — вершины, соединенные  $i$ -м ребром, а также его цена.

Выходные данные

Выведите одно число — цену максимального поддерева данного дерева.

входные данные
10 6 4 8 5 6 7 2 3 5 3 1 2 2 7 3 9 7 4 8 2 6 8 10 7 6 2 4
выходные данные
24

входные данные
6 1 3 12 5 3 4 3 4 2 2 4 5 6 2 6
выходные данные
12

7G. Мосты

1 секунда, 256 мегабайт

Санкт-Петербург расположен на  $n$  островах, которые соединены  $m$  мостами. Острова пронумерованы числами от 1 до  $n$ , а мосты — числами от 1 до  $m$ . Каждый мост соединяет два различных острова. Некоторые мосты были построены еще в эпоху Петра Великого, а некоторые построили недавно. Поэтому разные мосты имеют разную грузоподъемность. А именно, по  $i$ -му мосту могут ездить машины, масса которых не превышает  $d_i$ .

Вы разрабатываете проект, который должен помочь жителям города и туристам. Разработанный программный модуль должен отвечать на запросы: посчитать количество островов, на которые можно добраться с острова  $s_i$  на машине массы  $w_i$ , не превышая грузоподъемность мостов.

Входные данные

В первой строке записаны два числа  $n$  и  $m$  — количество островов и мостов в Санкт-Петербурге ( $1 \leq n \leq 10^5, 0 \leq m \leq 2 \cdot 10^5$ ).

Следующие  $i$  строк содержат по три числа:  $u_i, v_i$  и  $d_i$ , обозначающие, что  $i$ -й мост соединяет острова с номерами  $u_i$  и  $v_i$  и имеет грузоподъемность  $d_i$  ( $1 \leq u_i \neq v_i \leq n, 1 \leq d_i \leq 10^9$ ).

входные данные

3 3 7  
1 2  
2 3  
3 1  
ask 3 3  
cut 1 2  
ask 1 2  
cut 1 3  
ask 2 1  
cut 2 3  
ask 3 1

выходные данные

YES  
YES  
NO  
NO

7E. Школы

1 секунда, 256 мегабайт

С целью подготовки к проведению олимпиады по информатике мэр решил обеспечить надежным электроснабжением все школы города. Для этого необходимо провести линию электропередач от альтернативного источника электроэнергии "Майбутья" к одной из школ города (к какой неважно), а также соединить линиями электропередач некоторые школы между собой.

Считается, что школа имеет надежное электроснабжение, если она напрямую связана с источником "Майбутья", либо с одной из тех школ, которые имеют надежное электроснабжение.

Известна стоимость соединения между некоторыми парами школ. Мэр города решил выбрать одну из двух наиболее экономичных схем электроснабжения (стоимость схемы равняется сумме стоимостей соединений пар школ).

Напишите программу, которая вычисляет стоимость двух наиболее экономных схем альтернативного электроснабжения школ.

Входные данные

В первой строке входного файла находятся два натуральных числа, разделенных пробелом:  $N$  ( $3 \leq N \leq 100$ ), количество школ в городе, и  $M$  — количество возможных соединений между ними.

В каждой из последующих  $M$  строк находятся по три числа:  $A_i, B_i, C_i$ , разделенных пробелами, где  $C_i$  — стоимость прокладки линии электроснабжения ( $1 \leq C_i \leq 300$ ) от школы  $A_i$  до школы  $B_i$ .

Выходные данные

В единственной строке выходного файла должны содержаться два натуральных числа  $S_1$  и  $S_2$ , разделенных пробелом — две наименьшие стоимости схем ( $S_1 \leq S_2$ ).  $S_1 = S_2$  тогда и только тогда, когда существует несколько схем надежного электроснабжения наименьшей стоимости.

Гарантируется, что для входных данных существует две различные схемы надёжного электроснабжения.

входные данные

5 8  
1 3 75  
3 4 51  
2 4 19  
3 2 95  
2 5 42  
5 4 31  
1 2 9  
3 5 66

выходные данные

110 121

7F. Стоимость поддерева

В следующей строке записано число  $q$  ( $1 \leq q \leq 2 \cdot 10^5$ ) — количество запросов, на которые нужно ответить.

Каждая из следующих  $q$  строк содержит по два числа  $s_i$  и  $w_i$  ( $1 \leq s_i \leq n, 1 \leq w_i \leq 10^9$ ).

Выходные данные

Для каждого запроса выведите ответ в отдельной строке.

входные данные
7 8 1 2 3 1 6 7 2 3 2 2 7 5 3 4 2 4 5 2 5 6 4 6 7 1 7 1 6 1 2 2 2 2 4 4 2 1 1 1 3
выходные данные
2 7 7 2 7 7 5

7H. 0-1 MST

1 секунда, 256 мегабайт

Уджан накопил много ненужного хлама в своих ящиках, значительная часть которого является тетрадами с записями по математике: настало время их разобрать. Сейчас он нашёл старую запылившуюся тетрадь по теории графов с описанием одного графа.

Это неориентированный взвешенный граф с  $n$  вершинами. К тому же, это полный граф: каждая пара вершин соединена ребром. Вес каждого ребра равен либо 0, либо 1; к тому же, ровно  $m$  рёбер имеют вес 1, а все остальные рёбра имеют вес 0.

Так как Уджан не очень сильно желает разбирать свои записи, он решил найти вес минимального остовного дерева данного графа. (Вес остовного дерева графа равняется сумме весов всех его рёбер.) Можете ли вы найти ответ за Уджана, чтобы он прекратил валять дурака?

Входные данные

Первая строка ввода содержит два целых числа  $n$  и  $m$  ( $1 \leq n \leq 10^5, 0 \leq m \leq \min(\frac{n(n-1)}{2}, 10^5)$ ), количество вершин и количество рёбер веса 1 в данном графе.

$i$ -тая из следующих  $m$  строк содержит целые числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ), концы  $i$ -го ребра с весом 1.

Гарантируется, что ни одно ребро не повторяется во вводе.

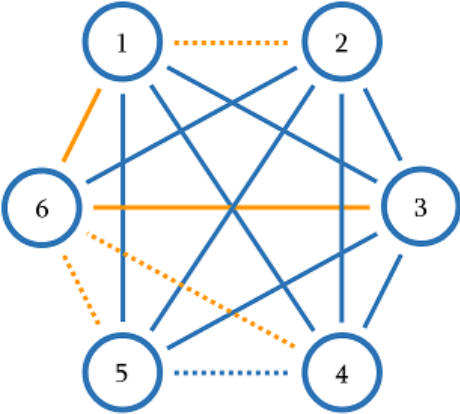
Выходные данные

Выведите одно целое число, вес минимального остовного дерева в данном графе.

входные данные
6 11 1 3 1 4 1 5 1 6 2 3 2 4 2 5 2 6 3 4 3 5 3 6
выходные данные
2

входные данные
3 0
выходные данные
0

Граф из первого примера показан на картинке ниже. Пунктирные рёбра имеют вес 0, все остальные рёбра имеют вес 1. Одно из возможных остовных деревьев покрашено в оранжевый цвет и имеет общий вес 2.



Во втором примере, вес каждого ребра 0, поэтому вес любого остовного дерева равен 0.

7I. Плотное остовное дерево

2 секунды, 256 мегабайт

Требуется найти в графе остовное дерево, в котором разница между весом максимального и минимального ребра минимальна.

Входные данные

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя целыми числами  $b_i, e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n, 0 \leq |w_i| \leq 10^9$ ).  $n \leq 1000, m \leq 10\,000$ .

Выходные данные

Если остовное дерево существует, выведите в первой строке выходного файла YES, а во второй строке одно целое число — минимальную разность между весом максимального и минимального ребра в остовном дереве.

В противном случае в единственной строке выведите NO.

входные данные
4 5 1 2 1 1 3 2 1 4 1 3 2 2 3 4 2

1 секунда, 256 мегабайт

Мальчик Кирилл написал однажды на листе бумаги строчку, состоящую из больших и маленьких латинских букв, а после этого ушел играть в футбол. Когда он вернулся, то обнаружил, что его друг Дима написал под его строкой еще одну строчку такой же длины. Дима утверждает, что свою строчку он получил циклическим сдвигом строки Кирилла на несколько шагов вправо (циклический сдвиг строки «abcde» на 2 позиции вправо даст строку «deabc»). Однако Дима известен тем, что может случайно ошибиться в большом количестве вычислений, поэтому Кирилл в растерянности – верить ли Диме? Помогите ему! По данным строкам выведите минимальный возможный размер сдвига или  $-1$ , если Дима ошибся.

Входные данные

Первые две строки входных данных содержат строки Кирилла и Димы, соответственно. Длины строк одинаковы, не превышают  $10^6$  и не равны 0.

Выходные данные

Выведите единственное число – ответ на вопрос задачи.

входные данные
zabcd abcdz
выходные данные
4

8Е. А-функция от строчки

1 секунда, 256 мегабайт

Дана строка  $S$ , состоящая из  $N$  символов. Определим функцию  $A(i)$  от первых  $i$  символов этой строки следующим образом:

$A(i) :=$  максимально возможному  $\{k\}$ , что равны следующие строки:

$S_1 + S_2 + \dots + S_k$  и  $S_i + S_{i-1} + \dots + S_{i-k+1}$ , где  $S_i - \{i\}$ -ый символ строки  $\{S\}$ , а знак  $+$  означает, что символы записываются в строчку непосредственно друг за другом.

Напишите программу, которая вычислит значения функции  $A$  для заданной строчки для всех возможных значений  $\{i\}$  от 1 до  $\{N\}$ .

Входные данные

В первой строке входного файла записано одно число  $N$ .  
 $1 \leq N \leq 200\,000$ . Во второй строке записана строка длиной  $N$  символов, состоящая только из больших и/или маленьких латинских букв.

Выходные данные

В выходной файл выведите  $N$  чисел – значения функции  $A(1), A(2), \dots, A(N)$ .

входные данные
5 aaba
выходные данные
1 2 0 1 5

8F. Поиск подмассива со сдвигом

1.5 секунд, 256 мегабайт

Даны два массива  $a$  и  $b$ , состоящие из натуральных чисел. Требуется найти все вхождения второго массива в первый с точностью до сдвига на какое-то целое число. Формально, корректным вхождением является индекс  $l$  такой, что для каждого индекса  $i \in [l, l + k)$   $a[l + i] = b[i] + d$ , где  $d$  - целое число, фиксированное для всех  $i$ .

Входные данные

выходные данные
YES 0

8А. Префикс-функция

1 секунда, 256 мегабайт

Вам дана строка из строчных латинских букв. От вас требуется вычислить префикс-функцию данной строки.

Входные данные

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает  $10^6$ .

Выходные данные

Требуется вывести префикс-функцию данной строки.

входные данные
abacaba
выходные данные
0 0 1 0 1 2 3

Предполагается, что значение префикс-функции для первого символа равно нулю.

8В. N-функция

1 секунда, 256 мегабайт

Вам дана строка из строчных латинских букв. От вас требуется вычислить N-функцию данной строки.

Входные данные

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает  $10^6$ .

Выходные данные

Требуется вывести N-функцию данной строки.

входные данные
abacaba
выходные данные
7 0 1 0 3 0 1

Предполагается, что значение N-функции для первого символа равно длине строки.

8С. Поиск подстроки (1 балл)

1 секунда, 256 мегабайт

Найти все вхождения строки  $T$  в строку  $S$ .

Входные данные

Первые две строки входных данных содержат строки  $S$  и  $T$ , соответственно. Длины строк больше 0 и меньше 500 000, строки содержат только строчные латинские буквы.

Выходные данные

Выведите номера символов, начиная с которых строка  $T$  входит в строку  $S$ , в порядке возрастания.

входные данные
ababbbababa aba
выходные данные
0 5 7

8D. Строчечки

В первой строке входного файла даны два числа  $n, k$  ( $1 \leq n, k \leq 10^5$ ) — длины первого и второго массивов соответственно. Во второй строке входного файла даны  $n$  чисел  $a_i$  через пробел, а в третьей строке —  $k$  чисел  $b_i$ , описывающие элементы массивов ( $0 \leq a_i, b_i \leq 10^9$ ).

Выходные данные

В единственной строке выходного файла выведите все индексы вхождений в возрастающем порядке. Числа требуется выводить через пробел.

входные данные
5 2 1 2 4 5 7 10 11
выходные данные
1 3

входные данные
10 3 1 2 1 2 1 10 10 15 16 15 1 2 1
выходные данные
1 3 8

8G. Потеряшка

1 s., 4 MB

Обратите внимание на необычные ограничения времени и памяти.

Герман потерялся в Интернете! Вы хотите узнать разбор этой задачи, поэтому Вам придется выручить его. Очевидно, что и Герман, и Интернет — это строки, состоящие из маленьких букв английского алфавита. Скажем, что Герман (обозначим соответствующую строку за  $G$ ) находится в Интернете (обозначим соответствующую строку за  $I$ ) в точке  $x$ , если выполнены равенства  $G_0 = I_x, G_1 = I_{x+1}, \dots, G_{|G|-1} = I_{x+|G|-1}$

Пусть Герман находится в Интернете в позициях  $x_1, x_2, \dots, x_t$ . Денис уже обнаружил Германа, но из принципа не будет его спасать до тех пор, пока Вы не сообщите Денису значение  $x_1 + x_2 + \dots + x_t$ .

Входные данные

Первая строка содержит число  $n$  ( $1 \leq n \leq 10^5$ ) — длину слова, ассоциированного с Германом.

Вторая строка содержит число  $m$  ( $1 \leq m \leq 67 \cdot 10^6$ ) — длину строки, представляющей из себя Интернет.

В третьей строке дано строковое представление Германа.

В четвертой строке дано строковое представление Интернета.

Выходные данные

Выведите одно число — сумму всех позиций, где можно обнаружить Германа в Интернете.

входные данные
6 19 german ogogermangegermange
выходные данные
14

В примере можно обнаружить Германа на позициях 3 и 11, поэтому ответом является 14.

9A. Мультимножество Василия

1 секунда, 256 мегабайт

Задачи - Codeforces

У автора уже закончились истории про Василия, поэтому он просто написал формальную постановку задачи.

У вас есть  $q$  запросов и мультимножество  $A$ , изначально содержащее только число 0. Запросы бывают трёх видов:

- «+ x» — добавить в мультимножество  $A$  число  $x$ .
- «- x» — удалить одно вхождение числа  $x$  из мультимножества  $A$ . Гарантируется, что хотя бы одно число  $x$  в этот момент присутствует в мультимножестве.
- «? x» — вам даётся число  $x$ , требуется вычислить  $\max_{y \in A} x \oplus y$ , то есть максимальное значение побитового исключающего ИЛИ (также известно как XOR) числа  $x$  и какого-нибудь числа  $y$  из мультимножества  $A$ .

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

Входные данные

В первой строке входных данных содержится число  $q$  ( $1 \leq q \leq 200\,000$ ) — количество запросов, которые требуется обработать Василию.

Каждая из последующих  $q$  строк входных данных содержит один трёх символов «+», «-» или «?» и число  $x_i$  ( $1 \leq x_i \leq 10^9$ ). Гарантируется, что во входных данных встречается хотя бы один запрос «?».

Обратите внимание, что число 0 всегда будет присутствовать в мультимножестве.

Выходные данные

На каждый запрос типа «?» выведите единственное целое число — максимальное значение побитового исключающего ИЛИ для числа  $x_i$  и какого-либо числа из мультимножества  $A$ .

входные данные
10 + 8 + 9 + 11 + 6 + 1 ? 3 - 8 ? 3 ? 8 ? 11
выходные данные
11 10 14 13

После первых пяти операций в мультимножестве  $A$  содержатся числа 0, 8, 9, 11, 6 и 1.

Ответом на шестой запрос будет число  $11 = 3 \oplus 8$  максимальное из чисел  $3 \oplus 0 = 3, 3 \oplus 9 = 10, 3 \oplus 11 = 8, 3 \oplus 6 = 5$  и  $3 \oplus 1 = 2$ .

9B. Библиотека

2 секунды, 512 мегабайт

Прошел почти год с момента, как Рик оказался на Флорине, однако его сознание никак не прояснялось. Воспоминания о прошлом были спрятаны в глубинах его разума, а может и вовсе утеряны. Однако сегодня что-то случилось. Рик вспомнил: у него была работа. Он анализировал Ничто. Наверное, Ничто — это космос, а значит Рик в прошлом был космоаналитиком. А еще Рик вспомнил, что все жители Флорины должны были погибнуть, но он не знал, почему.

Резидента Мирлина Теренса заинтересовала эта информация, поэтому он взял Рика с собой в библиотеку Верхнего города. Может быть, какая-нибудь литература по космоанализу могла бы вернуть ему память? Теренс не знал, что пропавшего космоаналитика активно ищут, а потому в библиотеке был получен приказ сообщать о любых посетителях, которые спросят о такой литературе. Библиотекарь отследил запросы наших героев в поисковой системе и поспешил вызвать патрульных.

Тем временем Теренс предложил Рiku ознакомиться с книгой известного автора Врийта "Трактат об инструментальном космоанализе". Рiku книга определенно показалась знакомой, особенно его привлекла строка  $s$ . Смысла самой строки, он, к сожалению, не понимал, однако в ее частях он видел что-то знакомое. Чтобы разобраться подробнее, Рик решил изучить все подстроки  $s$ . Однако изучать равные подстроки не было смысла, а остальные стоило как-либо систематизировать. Например, расставить их по длине и в алфавитном порядке. Поэтому Рик попросил вас узнать, сколько у данной строки существует пар подстрок  $s_1$  и  $s_2$  равной длины, таких, что  $s_1 < s_2$  лексикографически.

Входные данные

Задана строка  $s$ , состоящая из строчных латинских букв ( $|s| \leq 2500$ ).

Выходные данные

Выведите одно число — количество искомых пар подстрок.

входные данные
abac
выходные данные
9

Рассмотрим подстроки длины 1. Имеется две подстроки "a", каждая из которых меньше подстрок "b" и "c". Также подстрока "b" меньше подстроки "c". Отсюда получаем 5 пар искомых подстрок.

Теперь рассмотрим подстроки длины 2. Подстрока "ab" меньше подстрок "ba" и "ac", а строка "ac" меньше, чем строка "ba". Отсюда получаем еще 3 пары.

Наконец, рассмотрим подстроки длины 3. Подстрока "aba" меньше подстроки "bac".

Таким образом, суммарно получаем 9 искомых пар подстрок.

9C. К-я строка

1 секунда, 256 мегабайт

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку  $S$ ;
- найти в словаре  $k$ -ю строку в лексикографическом порядке.

Изначально словарь пуст.

Входные данные

Первая строка входного файла содержит число  $N$  — количество команд ( $1 \leq N \leq 10^5$ ).

Последующие  $N$  строк содержат по одной команде каждая. Команды записываются следующим образом:

- 1  $S$  — добавить строку  $S$  в словарь;
- 2  $k$  — вывести  $k$ -ю строку в лексикографическом порядке.

Гарантируется, что при запросе  $k$ -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает  $3 \cdot 10^5$ .

Все строки состоят из строчных латинских букв.

Выходные данные

Задачи - Codeforces

Для каждого запроса второго типа выведите  $k$ -ю в лексикографическом порядке строчку из словаря на момент запроса.

Гарантируется, что суммарная длина строк в выходном файле не превышает  $10^5$ .

входные данные
7 1 pushkin 1 lermontov 1 tolstoy 1 gogol 1 gorkiy 2 5 2 1
выходные данные
tolstoy gogol

9D. Сколько строк меньше

2 секунды, 512 мегабайт

Дан набор  $D$  из  $n$  строк и строка  $s$ . Требуется быстро находить число строк, лексикографически меньших  $s$ , в наборе  $D$ .

С заданной строкой  $s$  выполняются  $q$  модификаций, каждая из которых задается парой из числа  $k_i$  и символа  $c_i$ . Модификация  $(k_i, c_i)$  заключается в том, что все символы строки  $s$ , начиная с  $k_i$  и до конца строки, заменяются на символ  $c_i$ .

Например, пусть исходно строка  $s$  была равна «anatoly», тогда последовательность запросов (5, o), (3, b), (7, x) будет менять строку следующим образом:

«anatoly» → «anatooo» → «anbbbbbb» → «anbbbbbx»  
После каждого изменения строки  $s$  требуется вывести количество строк в наборе  $D$ , которые лексикографически меньше, чем  $s$ . Строка  $a$  лексикографически меньше строки  $b$ , если  $a \neq b$  и выполнено одно из двух условий:

- $a$  является префиксом строки  $b$ ;
- для некоторого  $i$  первые  $i$  символов строки  $a$  равны соответствующим символам строки  $b$ , а  $a_{i+1} < b_{i+1}$ .

Входные данные

Первая строка входных данных содержит два целых числа  $n$  и  $q$  — количество строк в наборе  $D$  и количество модификаций ( $1 \leq n, q \leq 10^6$ ).

Во второй строке находится строка  $s$ , состоящая из не более чем  $10^6$  строчных латинских букв.

В следующих  $n$  строках содержатся строки набора  $D$ . Каждая строка состоит из строчных латинских букв. Суммарная длина строк в  $D$  не превосходит  $10^6$ .

Следующие  $q$  строк содержат описания модификаций. Описание состоит из числа  $k_i$  и строчной буквы латинского алфавита  $c_i$ , разделенных пробелом ( $1 \leq k_i \leq |s|$ ).

Выходные данные

В первой строке выведите число строк в наборе  $D$ , которые лексикографически меньше начальной строки  $s$ .

Затем выведите  $q$  строк. В  $i$ -й строке выведите ответ после  $i$ -й модификации.



Все строки состоят из строчных латинских букв.

Выходные данные

Для каждой строки  $s_i$  выведите одно число: сколько раз она встречается в  $t$ .

входные данные
3 abc abcdr abcde xabcdef
выходные данные
1 0 1

9G. Бинарные Вирусы

1 секунда, 512 мегабайт

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой её подотрезок (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

Входные данные

Первая строка ввода содержит одно целое число  $n$ , равное количеству всех вирусных кодов. Каждая из следующих  $n$  строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30 000.

Выходные данные

Первая и единственная строка вывода должна содержать слово:

- ТАК — если бесконечная, безопасная последовательность из нулей и единиц существует;
- НIE — в противном случае.

входные данные
3 01 11 00000
выходные данные
NIE

входные данные
3 011 11 00000
выходные данные
ТАК

9H. Имя для ребенка

2 секунды, 512 мегабайт

входные данные
4 3 anatoly boris anatooo anbbbbbu anba 5 0 3 b 7 x
выходные данные
0 0 2 3

входные данные
5 5 abcde buz ababa build a aba 1 b 3 z 2 u 4 z 1 a
выходные данные
3 3 3 4 4 1

В первом тесте из примера строка изменяется следующим образом:

«anatoly» → «anatooo» → «anbbbbb» → «anbbbbx».

- Изначальная строка «anatoly» лексикографически меньше всех строк набора, поэтому ответ на задачу 0.
- После первого изменения строка становится «anatooo» и такая строка есть в наборе, однако ответ на задачу по прежнему будет 0, так как она не меньше, текущей.
- Затем строка становится «anbbbbb», что лексикографически больше, чем «anatooo» и «anba», но меньше чем «anbbbbbu» и «boris», таким образом ответ 2.
- После последнего изменения строка станет «anbbbbx», что лексикографически больше «anatooo», «anba» и «anbbbbbu», ответ 3.

Условие  
недоступно  
на  
русском  
языке

9F. Множественный поиск 2

2 секунды, 512 мегабайт

Дан массив строк  $S$  и строка  $t$ . Требуется для каждой строки  $p \in S$  определить, сколько раз она встречается в  $t$  как подстрока.

Входные данные

Первая строка входного файла содержит целое число  $n$  — число элементов в  $S$  ( $1 \leq n \leq 10^6$ ). Следующие  $n$  строк содержат по одной строке  $p \in S$ . Сумма длин всех строк из  $S$  не превосходит  $10^6$ . Последняя строка входного файла содержит  $t$  ( $1 \leq t \leq 10^6$ ).

9I. Знаки числа Пи

2 секунды, 256 мегабайт

Василий совсем недавно узнал об удивительных свойствах числа  $\pi$ . В одной из статей, которые он прочитал, была высказана гипотеза, что, какую бы последовательность цифр мы не взяли, рано или поздно эта последовательность встретится среди знаков числа  $\pi$ . Таким образом, если взять, например, роман «Война и мир», и закодировать его произвольным образом с помощью цифр, то мы обязательно найдем роман среди знаков числа  $\pi$ .

Василий был просто восхищен этим, поскольку это означает, что все книги, песни и программы уже написаны и зашифрованы в знаках числа  $\pi$ . Василия, конечно, немного насторожило то, что это только гипотеза, поэтому он решил проверить ее самостоятельно.

Для этого Василий скачал из интернета архив с последовательностью знаков числа  $\pi$  с некоторой позиции, и стал проверять разные строки из цифр на наличие в скачанном архиве. Короткие строки цифр Василий находил быстро, но когда он брал длинные строчки, оказывалось, что их в архиве нет. Василий решил взять за определение, что строка длины  $d$  встречается *наполовину*, если в ней есть подстрока длины хотя бы  $\lfloor \frac{d}{2} \rfloor$ , которая есть в архиве.

Для полноты картины Василий взял 2 больших числа  $x, y$  ( $x \leq y$ ), имеющих одинаковое количество цифр в десятичной записи, и теперь хочет найти количество чисел, находящихся в отрезке от  $x$  до  $y$ , которые встречаются в архиве *наполовину*. Помогите Василию вычислить остаток от деления этого значения на  $10^9 + 7$ .

Входные данные

В первой строке находится последовательность цифр  $s$  ( $1 \leq |s| \leq 1000$ ), в которой Василий собирается искать числа. Согласно гипотезе, такая последовательность цифр на самом деле встречается в десятичной записи числа  $\pi$ , однако мы не можем этого гарантировать.

Во второй и третьей строках находятся два положительных целых числа  $x, y$  одинаковой длины  $d$  ( $x \leq y, 2 \leq d \leq 50$ ). Числа  $x, y$  не содержат лидирующих нулей.

Выходные данные

Выведите количество чисел в отрезке от  $x$  до  $y$ , встречающихся *наполовину* в  $s$ , по модулю  $10^9 + 7$ .

входные данные
02 10 19
выходные данные
2

входные данные
023456789 10 19
выходные данные
9

входные данные
31415926535 10 29
выходные данные
20

У Джека родился ребенок, и они с женой хотят назвать придумать ему имя. Джек — математик, а его жена очень суеверна. Она хочет, чтобы имя ребенка содержало по крайней мере одно из имен ее предков в качестве подстроки. Они решили, что имя ребенка должно быть ровно длиной  $l$  символов, и должно состоять только из строчных латинских букв. Жена Джека дала Джеку список из  $n$  строк: имена ее предков. Джек находит все возможные имена для своего ребенка, и перечисляет их в алфавитном порядке. Он просит вас выбрать  $k$ -е возможное имя (нумерация с 1) в списке в качестве имени для своего ребенка.

Входные данные

Первая строка содержит числа  $l$  и  $k$ . Вторая строка содержит число  $n$ . Следующие  $n$  строк содержит имена предков  $s_i$ . ( $1 \leq L \leq 100, 1 \leq N \leq 100, 1 \leq s_i \leq 100, 1 \leq k \leq 10^{18}$ ).

Выходные данные

Выведите имя ребенка. Если требуемого имени не существует, выведите `unnamed baby` : (.

входные данные
5 10 2 like poke
выходные данные
epoke

входные данные
5 100 1 hack
выходные данные
unnamed baby :(

входные данные
3 5 3 ab bc ca
выходные данные
abd

входные данные
7 100000 4 abcd bcde cdef defg
выходные данные
dzdefgn

входные данные
10 1000000000 5 facebook hacker cup round three
выходные данные
ajpxcccuph