Содержание

Фурье и многочлены

Задача A0. Бесплатный плюсик [0.1 sec, 256 mb]

Даны многочлены $P(x)$ и $Q(x)$, найдите такие $A(x)$ и $R(x)$, что $P(x) = A(x)Q(x) + R(x)$ и $degR < degQ$. $Q(x) = x^k - 1$. У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m - 1$. Все вычисления происходят по модулю m, m — простое от 2 до $10^9 + 7$.

Формат входных данных

Первая строка содержит целые числа n, k, m.
Следующая строка содержит n чисел $p_0, p_1, \ldots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Формат выходных данных

На первой строке выведите $t = \max(n-k, 1)$ целых чисел от 0 до $m-1$: $a_0, a_1, \ldots, a_{t-1}$. На второй строке k целых чисел от 0 до $m-1$: $r_0, r_1, \ldots, r_{k-1}$. Все эти числа должны обладать свойством, что $P(x) = Q(x)(\sum a_i x^i) + \sum r_i x^i \mod m$.

Ограничения: $1 \leqslant n, k \leqslant 10^5$.

Примеры

stdin	stdout
3 1 7 1 5 1	6 1 0
3 1 7 2 4 1	5 1 0
3 1 7 0 4 1	5 1 5

Задача A1. FFT и манная кашка [0.2 sec, 256 mb]

Даны два многочлена
$A(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$
$B(x) = b_m x^m + b_{m-1} x^{m-1} + \cdots + b_0$
Найти $C(x) = A(x)B(x)$

Формат входных данных

$n, a_n, a_{n-1}, \ldots, a_0$
$m, b_m, b_{m-1}, \ldots, b_0$
$0 \leqslant n, m < 2^{16}$, $|a_i|, |b_j| \leqslant 9$
$a_n \neq 0, b_m \neq 0$

Формат выходных данных

Выведите коэффициенты C в том же формате.

Примеры

stdin	stdout
2 1 0 2 1 2 3	3 2 3 4 6

Задача A2. Деление многочленов [0.2 sec, 256 mb]

Даны два многочлена с коэффициентами из $\mathbb{Z}/7\mathbb{Z}$. Старший коэффициент обоих **не равен нулю**. Нужно поделить их с остатком.

Формат входных данных

Каждая из двух строк задаёт описание многочлена. Многочлен $a_k x^k + \cdots + a_2 x^2 + a_1 x + a_0$ описывается числом k ($0 \leqslant k \leqslant 50\,000$) и $k+1$ числами от 0 до 6: $a_k, \ldots, a_2, a_1, a_0$.

Формат выходных данных

На первой строке многочлен-частное. На второй строке многочлен-остаток. Выводите многочлены в том же формате. Если многочлен – тождественный ноль, для него $k = 0$.

Примеры

stdin	stdout
3 1 1 1 1 1 1 1	2 1 0 1 0 0
3 1 1 3 1 2 1 1 1	1 1 0 1 2 1
8 2 1 2 1 2 1 2 1 2 4 1 2 3 4 5	4 2 4 2 5 2 3 3 1 3 6

Автоматы и суффиксы

Задача B1. Суффиксный пулемёт [0.2 sec, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Теоретическая подготовка новобранцев армии Поссилтума включала в себя не только занятия по военному праву, но и начала криптографии. Лекции читал майор Мега Байт, не чуждый солдатского юмора. Гвидо и Нунцио, в чьё задание входил развал армии Поссилтума изнутри, решили на этом сыграть, внеся путаницу в терминологию. В начале очередной лекции Нунцио поднял руку и спросил:

— Вот вы на прошлой лекции рассказывали про конечные автоматы. А про конечные пулемёты расскажете?

Мега Байт не растерялся.

— Суффиксный пулемёт — это конечный автомат, принимающий все суффиксы данной строки (от нулевого до L-го включительно, где L — длина строки), и только их. Сержант Гвидо!

— Я, господин майор!

— Вы сможете отличить автомат от пулемёта?

— Так точно, господин майор!

— Вам дан конечный автомат. Требуется проверить, является ли он суффиксным пулемётом данной строки.

К сожалению, написание программ такого типа не входило в обязанности Гвидо и Нунцио как в Синдикате, так и в корпорации М. И. Ф. Так что соответствующую программу придётся писать Вам.

Это легенда по мотивам произведений Роберта Асприна,

Задача. Вам дан автомат и строка s. Проверьте, является ли данный автомат суффиксным пулемётом строки s, то есть, детерминированным конечным автоматом, принимающим ровно суффиксы строки s.

Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата N, количество переходов M, а также количество принимающих состояний T ($1 \leqslant T \leqslant N \leqslant 50\,000$, $1 \leqslant M \leqslant 100\,000$). Во второй строке через пробел заданы T различных чисел в пределах от 1 до N — принимающие состояния автомата, в возрастающем порядке. В последующих M строках заданы переходы в виде a_i b_i c_i, где $1 \leqslant a_i, b_i \leqslant n$, а c_i — маленькая буква латинского алфавита. Переход производится из состояния a_i в состояние b_i по букве c_i. Из каждого состояния a_i есть не более одного перехода по символу c_i. Последняя строка описания набора — это строка S, для которой автомат должен являться пулемётом. Она состоит только из маленьких латинских букв, и ее длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех N и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех M не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором $N = M = T = 0$.

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя формату примера.

Пример

stdin
2 1 2 1 2 1 2 a a 2 2 2 1 2 1 1 a 1 2 b ab 0 0 0
stdout
Automaton 1 is a machinegun. Automaton 2 is not a machinegun.

Задача B2. Суффиксный автомат [0.4 sec, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Дана строка. Постройте её суффиксный автомат.

Формат входных данных

Строка длины от 1 до 100 000, состоящая из маленьких латинских букв.

Формат выходных данных

На первой строке число состояний автомата и число рёбер. Следующие строки содержат рёбра в формате “откуда” “куда” “символ на ребре”. Далее число терминальных состояний и строка, содержащая все терминальные состояния в произвольном порядке. Начальным состоянием автомата должно быть состояние номер один.

Примеры

stdin	stdout
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

Задача B3. Помогите, спасите! [0.4 sec, 256 mb]

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

Формат входных данных

В единственной строке входных данных содержится непустая строка S, состоящая из N ($1 \le N \le 2 \cdot 10^5$) маленьких букв английского алфавита.

Формат выходных данных

Выведите N строк, в i-й строке должно содержаться количество различных подстрок в i-м префиксе строки S.

Примеры

stdin	stdout
aabab	1 2 5 8 11
atari	1 3 5 9 14

Задача B4. LZSS encoding [1 sec, 256 mb]

Алиса хочет отправить сообщение Бобу. Она хочет зашифровать сообщение, используя оригинальный метод шифрования. Сообщение – строка S, состоящая из N строчных английских букв.

$S[a \dots b]$ означает подстроку S от $S[a]$ до $S[b]$ $(0 \leqslant a \leqslant b < N)$. Если первые i букв уже зашифрованы, Алиса найдёт такие (j, k): $s[j..j+k] = s[i..i+k], k \geqslant 0, 0 \leqslant j < i, k = \max$. Если несколько j дают максимальное k, Алиса выберет минимальное j. Если $k > 0$ Алиса добавит пару $\langle j, k \rangle$ в шифр и увеличит i на k, иначе Алиса добавит -1 и ASCII код буквы $S[i]$ в шифр и увеличит i на 1.

Очевидно шифр начнёт с -1, далее будет ASCII код символа $S[0]$. Помогите Алисе реализовать её метод шифрования.

Формат входных данных

Перрвая строка ввода содержит количество тестов T $(1 \leqslant T \leqslant 50)$. Следующие T строк содержат сообщения для шифровки, каждое длины от 1 до 10^5, состоящие из строчных английских букв. Гарантируется, что суммарная длина всех сообщений не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого теста на отдельной строке выведите "`Case #X:`", где X – номер теста, нумерация с 1. Далее выведите шифр, в каждой строке по два целых числа через пробел.

Примеры

stdin	stdout
2 aaaaaa aaaaabbbbbaaabbc	Case #1: -1 97 5 0 Case #2: -1 97 4 0 -1 98 4 5 5 2 -1 99

Задача B5. Подстроки-4 [0.7 sec, 256 mb]

Даны K строк из маленьких латинских букв. Найдите их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leqslant K \leqslant 10$). Далее K строк длины от 1 до 200 000.

Формат выходных данных

Наибольшая общая подстрока.

Примеры

stdin	stdout
3 abacaba mycabarchive acabistrue	cab

Задача B6. Рефрен [0.4 sec, 256 mb]

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

По заданной последовательности требуется найти ее рефрен.

Формат входных данных

Два целых числа: n и m ($1 \leqslant n \leqslant 150\,000$, $1 \leqslant m \leqslant 10$).

Вторая строка содержит n целых чисел от 1 до m.

Формат выходных данных

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

Пример

stdin	stdout
9 3 1 2 1 2 1 3 1 2 1	9 3 1 2 1

Замечание

Эту задачу обязательно сдавать суффмассивом.

Даже если больше вам по душе деревья и автоматы.

Линейное программирование

Задача D1. Простая задача [0.1 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n+1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1+\ldots+c_nx_n$. Гарантируется, что решение существует и максимальное значение достигается.

Пример

stdin	stdout
2 2 1 2 3 2 1 3 1 1	2.0
1 1 1 2 -2	0.0
1 1 4 5 3	3.75

Задача D2. Простая задача [0.1 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Первая строка входного файла содержит два целых числа: n и m — количество переменных и количество уравнений ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n+1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле не превышают 100 по модулю.

Формат выходных данных

Выведите одно число: максимальное значение $c_1x_1+\ldots+c_nx_n$. Если решения нет, выведите "`No solution`". Если можно получить сколь угодно большое значение, выведите "`Unbounded`".

Пример

stdin	stdout
2 2 1 2 3 2 1 3 1 1	2.0
2 1 -1 -1 0 1 1	Unbounded
2 1 1 1 -1 1 1	No solution

Задача D3. Простая задача с хорошими тестами [0.1 sec, 256 mb]

Найдите оптимальное решение задачи линейного программирования.

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n \leqslant b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \leqslant b_m \\ x_1 \geqslant 0 \\ \ldots \\ x_n \geqslant 0 \\ c_1x_1 + \ldots + c_nx_n \to \max \end{cases}$$

Формат входных данных

Входной файл содержит один или несколько тестов. На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит два целых числа: n и m — количество переменных и количество неравенств ($1 \leqslant n, m \leqslant 5$). Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$. Следующая строка содержит n целых чисел: $c_1, \ldots, c_n$. Все числа во входном файле целые и не превышают 100 по модулю.

Формат выходных данных

Для каждого теста выведите одну строку.

Выведите максимальное значение $c_1x_1 + \ldots + c_nx_n$, пробел, двоеточие, пробел, сами числа $x_1x_2 \ldots x_n$. Все числа выводите с максимальной точностью. Если решения нет, выведите "`No solution`". Если можно получить сколь угодно большое значение, выведите "`Unbounded`".

Пример

stdin	stdout
4 2 2 1 2 3 2 1 3 1 1 2 1 -1 -1 0 1 1 2 1 1 1 -1 1 1 2 2 1 2 3 2 1 3 -10 1	2.0 : 1.0 1.0 Unbounded No solution 1.5 : 0 1.5

Задача D4. Быстрый симплекс [0.1 sec, 256 mb]

Дана системы равенств и неравенств:

$$\begin{cases} a_{11}x_1 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + \ldots + a_{2n}x_n = b_2 \\ \ldots \\ a_{m1}x_1 + \ldots + a_{mn}x_n = b_m \end{cases} \qquad \begin{cases} l_1 \leqslant x_1 \leqslant u_1 \\ \ldots \\ l_n \leqslant x_n \leqslant u_n \end{cases}$$

Также вам даны вектора $(c_{11}, c_{12}, \ldots, c_{1n}), \ldots, (c_{k1}, c_{k2}, \ldots, c_{kn})$.
$\forall i = 1..k$ найдите вектор x: $x_1c_{i1} + \cdots + x_nc_{in} \to \max$.

Формат входных данных

Входной файл содержит один или несколько тестов.

На первой строке число тестов, далее сами тесты. Каждый тест описывается следующим образом. Первая строка теста содержит три целых числа: n, m, k — количество переменных, количество уравнений, количество целевых функций ($1 \leqslant n, m, k \leqslant 30$).

Следующие m строк содержат по $n + 1$ целому числу: $a_{i1}, \ldots, a_{in}, b_i$.

Следующие n строк содержат пары чисел l_i, r_i ($l_i \leqslant r_i$).

Следующие k строк содержат по n целых чисел: $c_{i1}, \ldots, c_{in}$.

Все числа во входном файле целые и не превышают 100 по модулю.

Формат выходных данных

Для каждого теста выведите k строк, i-я строка – результат максимизации выражения $x_1c_{i1}+\cdots+x_nc_{in}$. Выведите максимальное значение $c_1x_1+\ldots+c_nx_n$, пробел, двоеточие, пробел, сами числа $x_1x_2\ldots x_n$. Все числа выводите с максимальной точностью. Гарантируется, что в каждом тесте решение существует. Ответы на тесты разделяйте пустой строкой.

Пример

stdin	stdout

Задача D5. Road times [0.2 sec, 256 mb]

Дорожная сеть страны – ориентированный граф. У каждого ребра есть длина, целое число от 1 до 1000, и ограничение на максимальную скорость, вещественное число от 30 до 60, в километрах в час. Длины вам известны, а ограничения – нет. Известно, что когда водитель такси берётся доставить пассажира из вершины a в вершину b, он осуществляет перевозку строго по кратчайшему пути между вершинами и едет с максимальной допустимой скоростью. Длина пути – сумма длин рёбер. Также известно, что $\forall a, b \; \exists$ *единственный* кратчайший путь из a в b.

Ваша задача – по длинам рёбер и уже сделанным поездкам a_i b_i $time_i$ оценить минимальное и максимальное время в пути между вершинами c_j d_j

Формат входных данных

На первой строке число вершин n ($1 \leqslant n \leqslant 30$).
Вершины нумеруются числами от 0 до $n - 1$.
Следующие n строк содержат матрицу $n \times n$ длин дорог. Дороги односторонние.
Отсутствие дорог обозначено числом -1, длины дорог целые от 1 до 1 000.
Всего не более 100 дорог.
Далее идёт число r ($1 \leqslant r \leqslant 100$) и r уже известных маршрутов a_i b_i $time_i$, $time_i \in \mathbb{R}$.
Времена в минутах (не в часах).
Затем число запросов q ($1 \leqslant q \leqslant 100$) и q строк c_j d_j.

Формат выходных данных

Для каждого запроса выведите четыре числа – откуда, куда, min время, max время.

Примеры

stdin	stdout
3 0 50 -1 55 0 40 -1 40 0 1 0 2 120 3 0 1 1 2 1 0	0 1 50.0 80.0 1 2 40.0 70.0 1 0 55.0 110.0