

Задача А. Суффиксный автомат

Имя входного файла: `automaton.in`
Имя выходного файла: `automaton.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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

<code>automaton.in</code>	<code>automaton.out</code>
<code>ababb</code>	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 В. Свобода выбора

Имя входного файла: `freedom.in`
Имя выходного файла: `freedom.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Даны две строки, состоящих из заглавных латинских букв. Нужно найти их наибольшую общую подстроку.

Формат входных данных

На первой строке число n ($1 \leq n \leq 10^5$).

На второй и третьей строках находятся по n заглавных английских букв.

Формат выходных данных

Максимальную по длине общую подстроку. Если оптимальных ответов несколько, выведите любой.

Примеры

<code>freedom.in</code>	<code>freedom.out</code>
28 VOTEFORTHEGREATALBANIAFORYOU CHOOSETHEGREATALBANIANFUTURE	THEGREATALBANIA

Задача С. Помогите, спасите!

Имя входного файла: `keepcounted.in`
Имя выходного файла: `keepcounted.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

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 ($1 \leq N \leq 2 \cdot 10^5$) 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

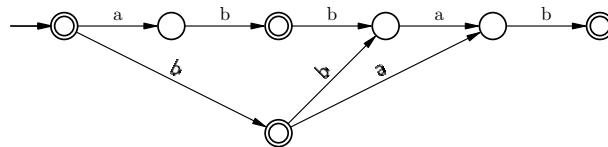
<code>keepcounted.in</code>	<code>keepcounted.out</code>
<code>aabab</code>	1 2 5 8 11
<code>atari</code>	1 3 5 9 14

Задача D. Суффиксный автомат

Имя входного файла: `suffix.in`
Имя выходного файла: `suffix.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Суффиксным автоматом для строки w называется детерминированный конечный автомат A , который допускает язык $\text{Suff}(w)$ — множество суффиксов слова w . Например, суффиксный автомат для слова $abbab$ должен допускать в точности следующие слова: $\{abbab, bbab, bab, ab, b, \varepsilon\}$. Мы также потребуем, чтобы суффиксный автомат не имел недостижимых состояний, и не было состояний, из которых не достижимы допустимые. Других ограничений, например, минимальности, накладывать не будем.

На рисунке показан суффиксный автомат для слова $abbab$.



По заданному скелету суффиксному автомату некоторого слова требуется восстановить суффиксный автомат. А именно — вам даны состояния, переходы, начальное состояние и допускающие состояния. Но пометки на ребрах удалены.

Вам следует расставить пометки на ребрах заданного суффиксного автомата, так чтобы он стал суффиксным автоматом некоторого слова w , а также найти это слово. Для простоты будем считать, что размер алфавита ничем не ограничен, вы можете использовать в качестве символов числа от 1 до k (k вы можете выбрать сами).

Формат входных данных

Первая строка входного файла содержит три целых числа: n , m и t — количество состояний, количество переходов, и количество допускающих состояний, соответственно ($2 \leq n \leq 200$, $1 \leq m \leq 1000$, $1 \leq t \leq n$). Вторая строка содержит t целых чисел — номера допускающих состояний (состояния пронумерованы с 1, начальное состояние имеет номер 1).

Следующие m строк описывают переходы: каждая строка содержит два целых числа s_i и t_i и описывает переходы из s_i в t_i .

Формат выходных данных

На первой строке выходного файла выведите два целых числа: l и k — длину слова w и размер алфавита. Используйте числа $\{1, \dots, k\}$ как элементы алфавита. k не должно превышать m .

Вторая строка должна содержать l целых чисел — слово w .

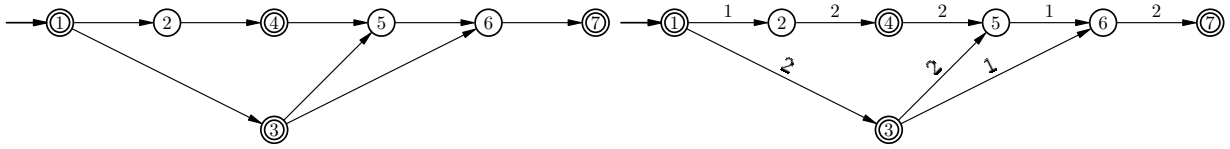
Наконец, третья строка должна содержать m целых чисел — метки на переходах скелета автомата, в том порядке, в котором они описаны во входном файле.

Гарантируется, что ответ всегда существует.

Примеры

suffix.in	suffix.out
7 8 4	5 2
1 3 4 7	1 2 2 1 2
1 2	1 2 2 2 1 2 1 2
1 3	
2 4	
3 5	
3 6	
4 5	
5 6	
6 7	

Замечание



Задача Е. Цензура

Имя входного файла: `censored.in`
Имя выходного файла: `censored.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Посчитайте, сколько строк над алфавитом из n символов длины m не содержат ни одной подстроки из заданного множества “запрещенных” строк.

Формат входных данных

В первой строке написаны целые числа n ($1 \leq n \leq 100$) — количество символов в алфавите, m ($1 \leq m \leq 100$) — длина искомых строк и p ($0 \leq p \leq 10$) — количество “запрещенных” подстрок. Следующая строка содержит n символов с кодами больше 32 — буквы алфавита. Далее идет p “запрещенных” строк, длины которых не превосходят $\min(m, 10)$ символов. Строки целиком состоят из символов алфавита.

Формат выходных данных

В первой строке выведите ответ на задачу.

Примеры

<code>censored.in</code>	<code>censored.out</code>
2 3 1 ab bb	5

Задача F. Суффиксный пулемёт

Имя входного файла:	suffix2.in
Имя выходного файла:	suffix2.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

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, внося 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.

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 ($1 \leq T \leq N \leq 50\,000$, $1 \leq M \leq 100\,000$). 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 $1 \leq a_i, b_i \leq n$, 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 её 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.

Примеры

suffix2.in	suffix2.out
2 1 2	Automaton 1 is a machinegun.
1 2	Automaton 2 is not a machinegun.
1 2 a	
a	
2 2 2	
1 2	
1 1 a	
1 2 b	
ab	
0 0 0	

Задача G. Две строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
Ограничение по памяти: 768 мегабайт

Даны две строки $S = S_0S_1 \cdots S_{|S|-1}$ и $T = T_0T_1 \cdots T_{|T|-1}$, каждая из которых состоит из маленьких латинских букв. Определим функцию $F(S, l, r)$ для строки S и подотрезка $[l, r]$ как

$$F(S, l, r) = r - l - \max(l, |S| - r - 1) + 1.$$

Другими словами, F обозначает длину подстроки, минус максимальное расстояние от подстроки до одной из границ строки S .

Ваша задача состоит в том, чтобы найти такую подстроку $S[l, r]$, что она встречается в T как подстрока, и значение $F(S, l, r)$ является максимально возможным.

Формат входных данных

Во входных данных содержатся строки S и T ($1 \leq |S|, |T| \leq 10^6$). Обе строки состоят из маленьких латинских букв.

Формат выходных данных

Выходные данные должны содержать два числа l и r такие, что $F(S, l, r)$ является максимально возможным ($0 \leq l \leq r < |S|$). В случае, если решений несколько, необходимо выбрать лексикографически минимальную пару (l, r) . Если ни одна из подстрок S не встречается в T , то оба числа должны быть равны -1.

Примеры

стандартный ввод	стандартный вывод
riveragesmalir toaxernaturaln	4 5
aaaaa aaaaa	0 4
amkar zenit	-1 -1

Задача Н. Несравнимые суффиксы

Имя входного файла: `incomparable.in`
Имя выходного файла: `incomparable.out`
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Путь задана некоторая строка s . Строка y называется *суффиксом* строки s в том случае, если $s = xy$ для некоторой строки x . Строка y называется *префиксом* строки s в том случае, если $s = yz$ для некоторой строки z .

Две строки a и b называются *различимыми* относительно s в том случае, если существует строка z такая, что az является суффиксом s , а bz не является суффиксом, или наоборот. К примеру, ab и b являются различимыми относительно $s = abbab$, так как $z = ab$ различает их: $abab$ не является суффиксом $abab$, а bab — является.

Два суффикса x и y строки s называются *несравнимыми* в том случае, если для любого префикса x_1 строки x и для любого префикса y_1 строки y строки x_1 и y_1 являются различимыми относительно s .

К примеру, суффиксы ab и bab строки $ababab$ являются несравнимыми, так как пары (a, b) , (a, ba) , (a, bab) , (ab, b) , (ab, ba) , (ab, bab) являются различимыми, а суффиксы bab и $abab$ не являются несравнимыми (так как ни одна строка не различает строки ba и $abba$), также суффиксы bab и $bbab$ не являются несравнимыми, так как их префиксы длины 1 равны b и, таким образом, не являются различимыми.

Дана строка s . Найдите такое подмножество S суффиксов этой строки, что все суффиксы S являются попарно несравнимыми, и при этом подмножество S имеет максимально возможный размер. Среди таких подмножеств необходимо выбрать то, сумма длин суффиксов в котором максимальна.

Формат входных данных

Входные данные содержат несколько тестовых примеров. Каждый тестовый пример состоит из единственной строки, содержащей исходную строку s . Каждая из строк является непустой и состоит из маленьких латинских букв. Суммарная длина всех строк во входных данных не превышает 10^5 , а общее число тестовых примеров не превосходит 1024.

Формат выходных данных

Для каждого тестового примера на первой строке выведите два числа: n — максимальное количество суффиксов в множестве S и l — максимальную суммарную длину строк в множестве S . Следующие n строк должны содержать сами суффиксы множества S в любом порядке, по одному на строке. В случае, если оптимальных ответов несколько, разрешается вывести любой из них.

Примеры

<code>incomparable.in</code>	<code>incomparable.out</code>
<code>ababab</code>	<code>2 6</code>
<code>aaaaa</code>	<code>ababab</code>
<code>abacaba</code>	<code>b</code>
	<code>1 5</code>
	<code>aaaaa</code>
	<code>3 7</code>
	<code>a</code>
	<code>ba</code>
	<code>caba</code>

Задача I. Суффиксный путь

Имя входного файла: `sufpath.in`
Имя выходного файла: `sufpath.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В одной супер-секретной лаборатории «Кимод йытысед» было разработано супер-секретное устройство. Синдикату «Черное солнце» удалось выяснить, что устройство имело супер-секретное название «Тамотва 173». Лучшие умы пытались расшифровать его, но так ничего и не удавалось. Пробовали и шифр цезаря, и сдвиг, и RSA. В конце концов, синдикат обратился за помощью в Весенний Гуманитарный Детский сад (ВГД). Как оказалось, это — также супер-секретная организация, которая занимается подготовкой элитного отряда дворников. Они профессионально умеют складывать мусор на самых видных местах, подметать так, что просыпается весь район и также немного увлекаются философией. Разумеется, они мгновенно поняли, что в названии зашифровано ни что иное, как основная часть этого устройства! Еще они выяснили, что это устройство умеет обрабатывать только файлы размера не более, чем шестьдесят мегабайт.

После этого синдикат обратился за помощью к лучшим друзьям ВГД, Ежедневному Женевскому Завтраку (ЕЖЗ). Они также специализируется на шифрах: красят заборы, моют полы, а также занимаются модернизацией и инновациями. ЕЖЗ сообщили, что это устройство — их профиль, оно идеально сочетается с их идеологией. Более формально, устройство также умеет выполнять модернизацию. К сожалению, с инновациями может справиться только ЕЖЗ, поэтому устройство не умеет их придумывать, а может лишь проверять, является ли некоторая инновация действительно революционной инновацией. Как оказалось, внутри устройства хранится строка, состоящая из маленьких латинских букв. Модернизация состоит в том, чтобы к хранящейся строке дописать маленькую латинскую букву. ЕЖЗ хотели сообщить более подробную информацию, но после фразы, что ВГД ошиблись в подсчете максимального размера обрабатываемого файла в тридцать раз, они были в непригодном для общения состоянии: у большинства отвалилась челюсть.

Синдикат обратился к своим последним друзьям: команде Инноваций и Культа Лени (ИКЛ). Они сразу объяснили, что строка является для устройства инновационной, если она является частью хранящейся в устройстве строки. Но всем сразу стало ясно: здесь есть какой-то подвох! И снова лучшие умы стали биться над этой задачей. Перепробовали всё: и кричать, и танцевать, и петь песенки, и бить баклуши. Но так им и не удавалось понять, что же происходит на самом деле. Наконец, директор ИКЛ пришёл к директору ЕЖЗ, и они, вместе с директором ВГД и мокренькой кисонькой, поняли, что строка должна быть не просто частью, а, будучи развернутой, должна являться префиксом развернутой строки, хранящейся в устройстве! Более того, она также должна состоять из маленьких латинских букв. После этого все вместе они пошли спать.

Из более достоверных источников (а именно Мадагаскарский Национальный Отряд Профессиональных Супер-агентов - МНОП) стало известно, что тот, у кого окажется данное устройство, получит неограниченную власть над миром. Ваша задача кристально ясна: реализуйте данное устройство.

Формат входных данных

В первой строке входного файла содержится число M — количество выполняемых операций. Далее в M строках содержатся описания операций: либо `+` `s` для модернизации, либо `?` `s` для проверки инновационности. Гарантируется, что устройство сможет обработать входной файл.

Формат выходных данных

Для каждого запроса проверки выведите «YES», если строка является инновационной, и «NO» в противном случае.

Примеры

<code>sufpath.in</code>	<code>sufpath.out</code>
2	YES
+ a	
? a	