

Задача А. Суффиксный автомат

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

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. Помогите, спасите!

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
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

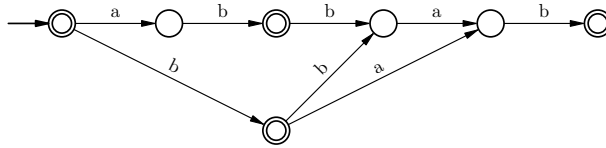
<code>stdin</code>	<code>stdout</code>
<code>aabab</code>	1 2 5 8 11
<code>atari</code>	1 3 5 9 14

Задача С. 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 допускающие. Других ограничений, например, минимальности, накладывать не будем.

На рисунке показан суффиксный автомат для слова *abbab*.



По заданному скелету суффиксного автомата некоторого слова требуется восстановить суффиксный автомат. A 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 .

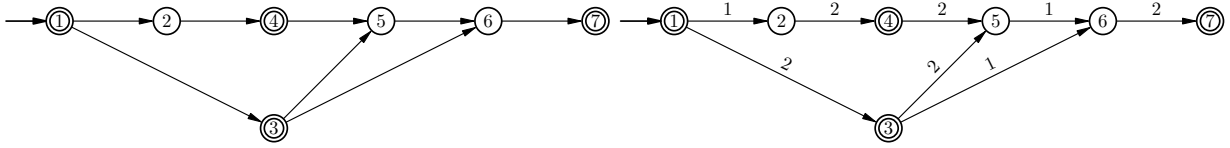
Наконец, третья строка должна содержать m целых чисел — метки на переходах скелета автомата, в том порядке, в котором они описаны во входном файле.

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 D. Ненокку

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входных данных

В каждой строчке входного файла записана одна из двух записей.

1. ? <слово> (<слово> — это набор не более 50 латинских символов): запрос проверки существования подстроки <слово> в произведении;
2. A <текст> (<текст> — это набор не более 10^5 латинских символов): добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более 10^5 символов. Суммарная длина всех запросов не превосходит 15 мегабайт плюс 12140 байт.

Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

Примеры

stdin	stdout
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Задача Е. 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 F. Различные подстроки

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

Дана строка S . Назовем ее подстрокой строку с i -го по j -й символ ($i \leq j$). Ваша задача — посчитать количество различных подстрок данной строки.

Формат входных данных

Во входном файле находится одна строка S , состоящая не более, чем из 200 000 символов. Все символы в строке — маленькие латинские буквы.

Формат выходных данных

В выходной файл выведите единственное число — количество различных подстрок заданной строки.

Примеры

стандартный ввод	стандартный вывод
aaba	8

Задача G. Рефрен

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

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

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 .

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.

Примеры

<code>refrain.in</code>	<code>refrain.out</code>
9 3	9
1 2 1 2 1 3 1 2 1	9
	1 2 1 2 1 3 1 2 1

Задача I. LZSS encoding

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

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

$S[a..b]$ означает подстроку S от $S[a]$ до $S[b]$ ($0 \leq a \leq b < N$). Если первые i букв уже зашифрованы, Алиса найдёт такие $(j, k) : s[j..j+k] = s[i..i+k], k \geq 0, 0 \leq j < i, k = \max$. 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

Первая строка ввода содержит количество тестов T ($1 \leq T \leq 50$). 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.

Пример

lzss.in	lzss.out
2	Case #1:
aaaaaa	-1 97
aaaaabbbbaaabbc	5 0
	Case #2:
	-1 97
	4 0
	-1 98
	4 5
	5 2
	-1 99

Замечание

Здесь и в следующих задачах запрещается использовать суффиксный массив для решения задач. Если вас переполняет ненависть к суффиксному дереву, можно использовать суфавтомат

Задача J. Общие подстроки

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

Дано K не обязательно различных строк из маленьких латинских букв, с суммарной длиной N . L_i определяется как максимальная длина строки, которая встречается как подстрока хотя бы у i строк из начального набора. Требуется для каждого $2 \leq i \leq K$ посчитать L_i .

Формат входных данных

В первой строке входных данных дано одно число L ($1 \leq L \leq 200\,000$) — число строк.

В следующих L строках даны сами строки из начального набора, по одной в строке. Гарантируется, что N — суммарная длина всех строк не превышает 200 000.

Формат выходных данных

В $k - 1$ строке выведите по одному числу — L_2, L_3, \dots, L_K .

Пример

стандартный ввод	стандартный вывод
6	5
matter	3
animate	2
pattern	2
thermal	1
domain	
teammate	