

Задача А. Суффиксный массив

Имя входного файла: `array.in`
Имя выходного файла: `array.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Постройте суффиксный массив для заданной строки s .

Формат входных данных

Единственная строка входного файла содержит строку s , состоящую из строчных латинских букв ($1 \leq |s| \leq 400\,000$).

Формат выходных данных

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

Примеры

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4
abacaba	7 5 1 3 6 2 4

Задача В. LCP для суффиксного массива

Имя входного файла: `sufflcp.in`
Имя выходного файла: `sufflcp.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана строка длины N и отсортированный массив суффиксов этой строки (т.е. суффиксный массив), вам нужно вычислить LCP. При сортировке строка `a` считается меньше строки `aa`. LCP — наибольший общий префикс двух последовательных суффиксов в суффиксном массиве.

Формат входных данных

В первой строке число N ($1 \leq N \leq 10^5$). На второй строке файла дана N строчных латинских букв. В третьей строке N чисел от 1 до N — суффиксный массив (числом i кодируется суффикс, начинающийся с i -го символа).

Формат выходных данных

Выведите $N - 1$ число — значения LCP.

Примеры

<code>sufflcp.in</code>	<code>sufflcp.out</code>
5	1
сасао	0
2 4 1 3 5	2
	0

Замечание

Суффиксный массив для строки `сасао`:

асао
ао
сасао
сао
о

Задача С. 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 D. 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 Е. Контрольное списывание

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

Сегодня на уроке преподаватель Массивов Автомат Укконеви́ч 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 Укконеви́ч дал им следующее задание: найти k -ю лексикографически непустую уникальную подстроку строки S .

Так как учитель знает, что Михаил В. и Роман Б. очень любят списывать у известного в узких кругах Максима И., каждый школьник получил своё число k и вынужден был обратиться к вам за помощью.

Формат входных данных

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.

Формат выходных данных

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

Примеры

<code>kthsubstr.in</code>	<code>kthsubstr.out</code>
<code>abacaba</code> <code>10</code>	<code>acab</code>
<code>abracadabra</code> <code>10000000000000000000</code>	<code>racadabra</code>

Задача F. Много строк

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

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

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$). В следующих K строках — собственно K строк (длины строк от 1 до 10 000).

Формат выходных данных

Наибольшая общая подстрока.

Примеры

<code>musubstr.in</code>	<code>musubstr.out</code>
3 abacaba mycabarchive acabistrue	cab

Задача Н. 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 I. Поиск подподстроки в подстроке

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4.5 секунд
Ограничение по памяти: 256 мегабайт

Опытным участникам соревнований по спортивному программированию хорошо известна классическая задача о нахождении количества вхождений подстроки в строку. Обычно она формулируется так: дана строка-образец s и строка t , требуется найти количество индексов, начиная с которых строка s содержится в строке t .

К сожалению, для решения этой задачи уже придумано множество алгоритмов, поэтому сама по себе она может быть интересна только в качестве упражнения, но не олимпиадной задачи. Однако, как это часто бывает со стандартными задачами, её легко усложнить — представим, что нас интересуют не сами строки s и t , а некоторые их подстроки $s[l_1 \dots r_1]$ и $t[l_2 \dots r_2]$.

Как вы уже, наверное, догадались, вам дано q запросов, i -й из которых задаёт некоторые подстроки $\bar{s} = s[l_{1i} \dots r_{1i}]$ и $\bar{t} = t[l_{2i} \dots r_{2i}]$. Для каждого такого запроса необходимо посчитать количество вхождений строки \bar{s} в строку \bar{t} .

Формат входных данных

Первая и вторая строки входных данных содержат строки s и t ($1 \leq |s|, |t| \leq 2 \cdot 10^5$) соответственно. Строки состоят из маленьких букв английского алфавита.

В третьей строке задано одно число q ($1 \leq q \leq 5 \cdot 10^5$) — количество запросов.

Каждая из следующих q строк содержит по четыре числа l_1, r_1, l_2 и r_2 ($1 \leq l_1 \leq r_1 \leq |s|, 1 \leq l_2 \leq r_2 \leq |t|$), описывающих очередной запрос.

Формат выходных данных

Выведите q чисел — ответы на запросы.

Примеры

стандартный ввод	стандартный вывод
abb	3
ababababb	1
5	2
1 2 1 7	1
2 3 2 9	4
3 3 4 7	
1 2 2 4	
1 1 1 9	

Задача J. Частотность

Имя входного файла: `frequent.in`
Имя выходного файла: `frequent.out`
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

В данный момент астробиологи ведут работу по изучению форм жизни на планете Альфабет. Жизнь там основана на ДНК, составленных из 26 различных нуклеотидов. Таким образом, ДНК каждой формы жизни на планете Альфабет может быть представлена строкой, состоящей из строчных букв английского алфавита. Астробиологи уже получили последовательности ДНК для K форм жизни, суммарная длина этих K последовательностей равняется N . Возможна ситуация, что ДНК некоторых форм жизни совпадают.

Теперь они хотели бы выделить некоторые нити (подстроки) данных ДНК, которые встречаются у различных форм жизни. Обозначим через $L(i)$ (здесь $2 \leq i \leq K$) максимальную длину нити (подстроки), состоящей из последовательных нуклеотидов, которая встречается хотя бы у i форм жизни. Обратите внимание, что $L(i)$ может быть равно 0.

Вычислите значения функции $L(i)$ для всех i от 2 до K .

Формат входных данных

В первой строке записано целое число K , означающее количество форм жизни, для которых была выделена последовательность ДНК. В каждой из последующих K строк записана непустая строка, состоящая из строчных букв английского алфавита. ($2 \leq N \leq 200\,000$, $2 \leq K \leq N$).

Формат выходных данных

Выходной файл должен содержать $K - 1$ строку со значениями $L(2), L(3), \dots, L(K)$, каждое на отдельной строке.

Примеры

<code>frequent.in</code>	<code>frequent.out</code>
6	5
<code>matter</code>	3
<code>animate</code>	2
<code>pattern</code>	2
<code>thermal</code>	1
<code>domain</code>	
<code>teammate</code>	