

Задача А. Суффиксное дерево

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

Дана строка s . Постройте сжатое суффиксное дерево для строки s и выведите его. Найдите такое дерево, которое содержит минимальное количество вершин.

Формат входных данных

В первой строке записана строка s ($1 \leq |s| \leq 10^5$), последний символ строки доллар «\$», остальные символы строки маленькие латинские буквы.

Формат выходных данных

Пронумеруйте вершины дерева от 0 до $n - 1$ в порядке обхода в глубину, обходя поддеревья в порядке лексикографической сортировки исходящих из вершины рёбер. Используйте ASCII-коды символов для определения их порядка.

В первой строке выведите число n – количество вершин дерева. В следующих $n - 1$ строках выведите описание вершин дерева, кроме корня, в порядке увеличения их номеров.

Описание вершины дерева v состоит из трёх целых чисел: p, lf, rf , где p ($0 \leq p \leq n, p \neq v$) – номер родителя текущей вершины. На ребер ведущем из p в v написана подстрока $s[lf..rf)$ ($0 \leq lf < rf \leq |s|$).

Примеры

suftree.in	suftree.out
aaa\$	7 0 3 4 0 0 1 2 3 4 2 1 2 4 3 4 4 2 4
b\$	3 0 1 2 0 0 2
ababa\$	10 0 5 6 0 0 1 2 5 6 2 1 3 4 5 6 4 3 6 0 1 3 7 5 6 7 3 6

Замечание

В этой задаче запрещается строить суфдерево через другие структуры данных.

Задача В. 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 С. Шпион

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Спецслужбы обнаружили действующего иностранного агента. Шпиона то есть. Установили наблюдение и выяснили, что каждую неделю он через Интернет посылает кому-то странные нечитаемые тексты. Чтобы выяснить, к какой информации получил доступ шпион, требуется расшифровать информацию. Сотрудники спецслужб проникли в квартиру разведчика, изучили шифрующее устройство и выяснили принцип его работы.

На вход устройства подается строка текста $S_1 = s_1s_2 \dots s_N$. Получив ее, устройство строит все циклические перестановки этой строки, то есть $S_2 = s_2s_3 \dots s_Ns_1, \dots, S_N = s_Ns_1s_2 \dots s_{N-1}$. Затем множество строк S_1, S_2, \dots, S_N сортируется лексикографически по возрастанию. И в этом порядке строчки выписываются в столбец, одна под другой. Получается таблица размером NN . В какой-то строке K этой таблицы находится исходное слово. Номер этой строки вместе с последним столбцом устройство и выдает на выход. Например, если исходное слово $S_1 = \text{abracadabra}$, то таблица имеет такой вид:

1. $\text{aabracadabr} = S_{11}$
2. $\text{abraabracad} = S_8$
3. $\text{abracadabra} = S_1$
4. $\text{acadabraabr} = S_4$
5. $\text{adabraabrac} = S_6$
6. $\text{braabracada} = S_9$
7. $\text{bracadabraa} = S_2$
8. $\text{cadabraabra} = S_5$
9. $\text{dabraabraca} = S_7$
10. $\text{raabracadab} = S_{10}$
11. $\text{racadabraab} = S_3$

И результатом работы устройства является число 3 и строка rdarcaaaabb . Это все, что известно про шифрующее устройство. А вот дешифрующего устройства не нашли. Но поскольку заведомо известно, что декодировать информацию можно (а иначе зачем же ее передавать?), Вам предложили помочь в борьбе с хищениями секретов и придумать алгоритм для дешифровки сообщений. А заодно и реализовать дешифратор.

Формат входных данных

В первой и второй строках находятся соответственно целое число и строка, возвращаемые шифратором. Длина строки и число не превосходят 100000. Строка содержит лишь следующие символы: a-z, A-Z. Других символов в строке нет. Лексикографический порядок на множестве слов задается таким порядком символов:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz

Символы здесь выписаны в порядке возрастания.

Формат выходных данных

Выведите декодированное сообщение в единственной строке.

Пример

стандартный ввод	стандартный вывод
3 rdarcaaaabb	abracadabra

Задача D. 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>	<code>7 9</code> <code>1 2 a</code> <code>1 7 b</code> <code>2 3 b</code> <code>3 4 a</code> <code>3 6 b</code> <code>4 5 b</code> <code>5 6 b</code> <code>7 4 a</code> <code>7 6 b</code> <code>3</code> <code>6 7 1</code>

Задача Е. 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 F. 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 G. Пруфффикс

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

Даны две строки $s = s_1s_2 \dots s_n$ и $t = t_1t_2 \dots t_m$. Даны несколько запросов вида (l_s, r_s, l_t, r_t) , и для каждого запроса нужно посчитать число пар (x, y) таких, что

- $l_s \leq x \leq r_s$,
- $l_t \leq y \leq r_t$ и
- строка $s_x s_{x+1} \dots s_n t_1 t_2 \dots t_y$ является подстрокой s или t .

Формат входных данных

Во входных данных содержится несколько тестов. Для каждого теста:

Первая строка содержит три целых числа n , m и q , длины s , t и количество запросов соответственно ($1 \leq n, m, q \leq 5 \times 10^5$).

Вторая строка содержит строку s длины n . Третья строка содержит строку t длины m . Обе строки состоят из маленьких латинских букв.

Каждая из следующих q строк содержит четыре целых числа l_s, r_s, l_t, r_t обозначающих запрос ($1 \leq l_s \leq r_s \leq n, 1 \leq l_t \leq r_t \leq m$).

Сумма всех значений n во всех тестах во входных данных не превышает 5×10^5 .

Сумма всех значений m во всех тестах во входных данных не превышает 5×10^5 .

Сумма всех значений q во всех тестах во входных данных не превышает 5×10^5 .

Формат выходных данных

Выведите одно целое число для каждого запроса, являющееся ответом на запрос.

Примеры

стандартный ввод	стандартный вывод
3 3 3	3
aaa	0
aaa	1
1 3 1 3	
1 1 2 2	
3 3 1 1	