

Содержание

Must have	2
Задача 22А. Суффиксный массив [0.15, 256]	2
Задача 22В. Циклические суффиксы [0.1, 256]	3
Задачи здорового человека	4
Задача 22С. Преобразование строковых функций [0.1, 256]	4
Задача 22D. Поиск набора образцов [0.1, 256]	5
Задача 22Е. Суффиксный массив (3 балла) [0.4, 256]	6
Для искателей острых ощущений	7
Задача 22F. Вирусы [0.1, 256]	7
Задача 22G. LZSS encoding [1.5, 256]	8

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

Обратите внимание на GNU C++ компиляторы с суффиксом inc.

Подни можно пользоваться **дополнительной библиотекой** (optimization.h).

То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.

И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): **пример**.

Для тех, кто хочет разобраться, как всё это работает.

Короткая версия быстрого ввода-вывода (**тык**) и короткая версия аллокатора (**тык**).

Must have

Задача 22А. Суффиксный массив [0.15, 256]

Данна строка, требуется построить суффиксный массив для этой строки. Суффиксный массив — лексикографически отсортированный массив всех суффиксов строки. Каждый суффикс задается целым числом — позицией начала.

Строка s лексикографически меньше строки t , если есть такое i , что $s_i < t_i$ и $s_j = t_j$ для всех $j < i$. Или, если такого i не существует и строка s короче строки t .

Здесь s_i — код i -го символа строки s .

Формат входных данных

Файл состоит из единственной строки. Эта строка — **английский литературный текст**. Длина текста не превосходит 10^5 . Коды всех символов в тексте от 32 до 127.

Формат выходных данных

Выведите N чисел — суффиксный массив данной строки.

Пример

stdin	stdout
99 bottles of beer.	14 3 11 19 2 1 15 4 16 17 9 13 8 12 5 18 10 7 6

Замечание

Это **очень** простая задача.

Предполагается решение за $\mathcal{O}(n \log n \cdot \text{time}(\text{less}))$.

Нормальное время написания — 5-10 минут.

Задача 22В. Циклические суффиксы [0.1, 256]

Рассмотрим строку $S = s_1s_2s_3 \dots s_{n-1}s_n$ над алфавитом Σ . Циклическим расширением порядка t строки S назовем строку $s_1s_2s_3 \dots s_{n-1}s_ns_1s_2 \dots$ из t символов; это значит, что мы приписываем строку S саму к себе, пока не получим требуемую длину, и берем префикс длины t .

Циклической строкой \tilde{S} назовем бесконечное циклическое расширение строки S .

Рассмотрим суффиксы циклической строки \tilde{S} . Очевидно, существует не более $|S|$ различных суффиксов: $(n + 1)$ -ый суффикс совпадает с первым, $(n + 2)$ -ой совпадает со вторым, и так далее. Более того, различных суффиксов может быть даже меньше. Например, если $S = abab$, первые четыре суффикса циклической строки \tilde{S} — это:

$$\begin{aligned}\tilde{S}_1 &= ababababab\dots \\ \tilde{S}_2 &= bababababa\dots \\ \tilde{S}_3 &= ababababab\dots \\ \tilde{S}_4 &= bababababa\dots\end{aligned}$$

Здесь существует всего два различных суффикса, в то время как $|S| = 4$.

Отсортируем первые $|S|$ суффиксов \tilde{S} лексикографически. Если два суффикса совпадают, первым поставим суффикс с меньшим индексом. Теперь нас интересует следующий вопрос: на каком месте в этом списке стоит сама строка \tilde{S} ?

Например, рассмотрим строку $S = cabcab$:

$$\begin{aligned}(1) \quad \tilde{S}_2 &= abcabcabca\dots \\ (2) \quad \tilde{S}_5 &= abcabcabca\dots \\ (3) \quad \tilde{S}_3 &= bcabcabcab\dots \\ (4) \quad \tilde{S}_6 &= bcabcabcab\dots \\ (5) \quad \tilde{S}_1 &= cabcabcab\dots \\ (6) \quad \tilde{S}_4 &= cabcabcab\dots\end{aligned}$$

Здесь циклическая строка $\tilde{S} = \tilde{S}_1$ находится на пятом месте.

Вам дана строка S . Ваша задача — найти позицию циклической строки \tilde{S} в описанном порядке.

Формат входных данных

Во входном файле записана единственная строка S ($1 \leq |S| \leq 1\,000\,000$), состоящая из прописных латинских букв.

Формат выходных данных

В выходной файл выведите единственное число — номер строки \tilde{S} в описанном порядке среди первых $|S|$ суффиксов.

Примеры

stdin	stdout
abracadabra	3
cabcab	5

Замечание

Разобрано на практике. Z-функция.

Задачи здорового человека

Задача 22С. Преобразование строковых функций [0.1, 256]

Для строки S определим Z -функцию следующим образом: $Z[i] = lcp(S, S[i..|S|])$, где $lcp(S_1, S_2)$ равно длине наибольшего общего префикса строк S_1 и S_2 . Например, для $S = abacabaa$ Z -функция равна $[8, 0, 1, 0, 3, 0, 1, 1]$.

Для строки S определим ее префикс-функцию: $\pi[i] = \max\{k | 0 \leq k < i, S[1..k] = S[i-k+1..i]\}$. Например, для $S = abacabaa$ ее префикс-функция имеет вид: $[0, 0, 1, 0, 1, 2, 3, 1]$.

Для некоторой строки S была посчитана ее Z -функция, а строка S была утеряна. Ваша задача получить ее префикс-функцию по заданной Z -функции.

Формат входных данных

В первой строке входного файла содержится натуральное число N ($1 \leq N \leq 200\,000$), где N — длина S . Во второй строке записана Z -функция строки S .

Формат выходных данных

Выведите N чисел — искомую префикс-функцию.

Пример

stdin	stdout
8	0 0 1 0 1 2 3 1
8 0 1 0 3 0 1 1	

Подсказка по решению

Очень простая задача. Решается буквально 1-2 циклами. На понимание Z , prefix.

Задача 22D. Поиск набора образцов [0.1, 256]

Напишите программу, которая для каждой строки из заданного набора S проверяет, верно ли, что она содержит как подстроку одну из строк из набора T .

Формат входных данных

Первая строка входного файла содержит натуральное число n — количество строк в наборе T ($n \leq 1000$). Каждая из следующих n строк содержит непустую строку длины не более 80-ти символов.

Оставшаяся часть файла содержит строки из набора S . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой и ее длина не превышает 250-ти символов.

Гарантируется, что размер входного файла не превышает 1 Мбайт.

Формат выходных данных

В выходной файл выведите все строки из набора S (в том порядке, в котором они находятся во входном файле), содержащие как подстроку хотя бы одну строку из набора T .

Пример

stdin	stdout
3 gr sud abc lksh sudislavl kostroma summer group a'	sudislavl group a'

Подсказка по решению

Собственно Ахо-Корасик.

Если у вас WA 36, вы неправильно читаете входные данные. Строки могут состоять только из пробелов.

Задача 22Е. Суффиксный массив (3 балла) [0.4, 256]

Постройте суффиксный массив для заданной строки s .

Формат входных данных

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

Формат выходных данных

Выведите такую перестановку начальных позиций суффиксов, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

Примеры

stdin	stdout
ababb	1 3 5 2 4

Замечание

ababb
abb
b
babb
bb

Подсказка по решению

А вот и он!

Для искателей острых ощущений

Задача 22F. Вирусы [0.1, 256]

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее сегмент (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

Пример

Для множества кодов {011, 11, 0000} примером бесконечной безопасной последовательности является 010101... Для множества {01, 11, 00000} бесконечной безопасной последовательности не существует.

Формат входных данных

Первая строка входного файла `virus.in` содержит одно целое число N , равное количеству всех вирусных кодов. Каждая из следующих n строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30 000.

Формат выходных данных

Первая и единственная строка выходного файла должна содержать слово:

- ТАК — если бесконечная, безопасная последовательность из нулей и единиц существует;
- NIE — в противном случае.

Пример

stdin	stdout
3 01 11 00000	NIE
3 011 11 0000	ТАК

Подсказка по решению

Задача на автомате, полученном по ходу Ахо-Корасик.

Задача 22G. LZSS encoding [1.5, 256]

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 $\langle k, j \rangle$ 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.

Примеры

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

Подсказка по решению

Предполагается решение суффмассивом + `set<int>` для поиска ближайшего слева-справа меньшего.