

Содержание

Must have	2
Задача 9А. Суффиксный массив [0.2 sec, 256 mb]	2
Задача 9В. LCP для суффиксного массива [0.3 sec, 256 mb]	3
Обязательные задачи	4
Задача 9С. Суффиксный массив [1 sec, 256 mb]	4
Задача 9D. Поиск набора образцов [0.4 sec, 256 mb]	5
Дополнительные задачи	6
Задача 9Е. Вирусы [0.2 sec, 256 mb]	6
Задача 9F. Набор строк [3.5 sec, 256 mb]	7

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

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

Обратите внимание на GNU C++ компиляторы с суффиксом inc, они позволяют пользоваться **дополнительной библиотекой**. Под ними можно сдать **вот это**.

Must have

Задача 9А. Суффиксный массив [0.2 sec, 256 mb]

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

Подсказка по решению

Это очень простая задача. Простая именно потому, что текст *литературный*.

Задача 9B. LCP для суффиксного массива [0.3 сек, 256 mb]

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.

Пример

stdin	stdout
5 сасао 2 4 1 3 5	1 0 2 0

Замечание

Суффиксный массив для строки `сасао`:

асао
ао
сасао
сао
о

Подсказка по решению

24-й тест — антихеш тест.

Можно сдать за $\mathcal{O}(n \log n)$, можно за $\mathcal{O}(n)$.

Обязательные задачи

Задача 9С. Суффиксный массив [1 сек, 256 mb]

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

Вот и он, вот и он! Долгожданный...

Задача 9D. Поиск набора образцов [0.4 sec, 256 mb]

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'

Замечание

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

Дополнительные задачи

Задача 9E. Вирусы [0.2 sec, 256 mb]

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;
- **НИЕ** — в противном случае.

Пример

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

Задача 9F. Набор строк [3.5 сек, 256 mb]

В Инновационном Отделе НИИ Исследований Данных Строк разработана клавиатура для внутреннего пользования, облегчающая набор строк огромной длины. Кроме обычных клавиш, соответствующих маленьким латинским буквам, на клавиатуре есть еще n функциональных клавиш F_1, \dots, F_n , соответствующих заданным строкам из словаря S_1, \dots, S_n . При нажатии такой клавиши F_i строка S_i загружается во внутреннюю память клавиатуры. В каждый момент времени в памяти может находиться не более одной строки из словаря.

Кроме того, в клавиатуру встроен графический манипулятор «Кыш», с помощью которого легким движением руки можно ввести любую подстроку находящейся в памяти строки.

Вася занимается исследованием эффективности данного нововведения. Для этого ему требуется написать программу, которая будет вычислять минимальное необходимое количество действий (нажатий и использований «Кыш») для ввода данной строки S . В момент начала ввода строки память пуста.

Например, если требуется ввести строку «abacaba», а в словаре есть строки «baba» и «саса», то это можно сделать за четыре действия — нажать F_1 , выбрать манипулятором подстроку «aba», затем нажать ‘с’, и опять выбрать манипулятором подстроку «aba». Если бы нужно было набрать с таким словарем «bacababa», то это можно сделать за пять действий: ‘b’, F_2 , «аса», F_1 , «baba».

Формат входных данных

В первой строке входного файла задано число n ($1 \leq n \leq 50$). В последующих n строках заданы S_i , составленные из не более чем 500 символов. В последней строке вводится непустая строка S , длина которой не превосходит 100 000. Все символы строк — маленькие латинские буквы.

Формат выходных данных

Выведите минимальное необходимое количество действий.

Пример

stdin	stdout
2 baba саса abacaba	4
2 baba саса bacababa	5

Замечание

OK можно без суффдерева, обычным алгоритмом Ахо-Корасик.

C суффиксным деревом/автоматом работать будет, конечно, ещё быстрее.