

Содержание

Must have	2
Задача 8А. Словарь [0.7 sec, 256 mb]	2
Задача 8В. К-я строка [0.3 sec, 256 mb]	3
Обязательные задачи	4
Задача 8С. Общая подстрока [2.3 sec, 256 mb]	4
Задача 8D. Циклические сдвиги [0.3 sec, 256 mb]	5
Задача 8Е. Свобода выбора [1 sec, 256 mb]	6
Дополнительные задачи	7
Задача 8F. Подстроки-3 [0.2 sec, 256 mb]	7
Задача 8G. Рефрен [1.5 sec, 256 mb]	8

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 8А. Словарь [0.7 sec, 256 mb]

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

Формат входных данных

В первой строке дан текст (не более 10^6 строчных латинских букв).

Далее дано число M — количество слов в словаре.

В следующих M строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более 10^5 .

Формат выходных данных

M строк вида Yes, если слово присутствует, и No иначе.

Пример

stdin	stdout
trololo	No
3	Yes
abacabadabacaba	Yes
olo	
trol	

Замечание

Эта задача решается **просто бором**. Без алгоритма Ахо-Корасик.

Задача 8В. К-я строка [0.3 sec, 256 mb]

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

stdin	stdout
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

Подсказка по решению

Бор.

Обязательные задачи

Задача 8С. Общая подстрока [2.3 sec, 256 mb]

Заданы две строки, состоящие из 0 и 1. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них k -ую в лексикографическом порядке.

Строка S меньше строки T в лексикографическом порядке, если выполняется одно из двух условий:

- S является префиксом T ;
- существует i , не превышающее длин строк S и T , такое, что для $j < i$ выполняется $S[j] = T[j]$ и $S[i] < T[i]$.

Формат входных данных

Первые две строки входного файла содержат заданные строки, длиной не более 4000 символов каждая. Третья строка содержит целое положительное число k , не превышающее количества общих подстрок двух заданных строк.

Формат выходных данных

Выведите в выходной файл k -ую в лексикографическом порядке общую подстроку заданных строк.

Пример

stdin	stdout
0100 0010 3	01

Подсказка по решению

Несжатый бор. Вернее даже суффиксное дерево.

Задача 8D. Циклические сдвиги [0.3 sec, 256 mb]

k -м циклическим сдвигом строки S называется строка, полученная перестановкой k первых символов строки S в конец строки.

Рассмотрим все различные циклические сдвиги строки S и отсортируем их по возрастанию. Требуется вычислить i -ю строчку этого массива.

Например, для строки `abacabac` существует четыре различных циклических сдвига: нулевой (`abacabac`), первый (`bacabaca`), второй (`acabacab`) и третий (`cabacaba`). После сортировки по возрастанию получится такой массив: `abacabac`, `acabacab`, `bacabaca`, `cabacaba`.

Формат входных данных

В первой строке входного файла записана строка S , длиной не более 100 000 символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число k ($1 \leq k \leq 100\,000$).

Формат выходных данных

В выходной файл выведите k -й по возрастанию циклический сдвиг строки S , или слово IMPOSSIBLE, если такого сдвига не существует.

Пример

stdin	stdout
abacabac 4	cabacaba
abacabac 5	IMPOSSIBLE

Замечание

Эта задача всё ещё проще чем «построить суффмассив за $\mathcal{O}(n \log n)$ ».

Кстати, в ней нет антихештестов. Это значит, что хеши по модулю 2^{64} работают также хорошо, как по простому, близкому к 2^{64} ;-)

Задача 8E. Свобода выбора [1 sec, 256 mb]

Даны две строки, состоящих из заглавных латинских букв. Нужно найти их наибольшую общую подстроку. Полное условие можно посмотреть на [тимусе](#).

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

stdin	stdout
28 VOTEFORTHEGREATALBANIAFORYOU CHOOSETHEGREATALBANIANFUTURE	THEGREATALBANIA

Замечание

19-й — антихеш тест.

Разбиралась на лекции. Здесь всё ещё не нужны суффмассивы.

Дополнительные задачи

Задача 8F. Подстроки-3 [0.2 sec, 256 mb]

Даны K строк из маленьких латинских букв. Найдите 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

stdin	stdout
3 abacaba mucabarchive acabistrue	cab

Замечание

Нужно написать хороший код, чтобы пройти ТЛ.

В частности вам точно понадобится рукописная хеш-таблица с открытой адресацией.

Задача 8G. Рефрен [1.5 sec, 256 mb]

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

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.

Пример

stdin	stdout
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Замечание

Эту задачу обязательно сдавать *суффмассивом*.

Даже если вам больше по душе деревья и автоматы.