Содержание

Общая информация:

Вход в контест: `http://contest.yandex.ru/contest/1145/`
Дедлайн на задачи: 2 недели, до 5-го апреля 23:59.
К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).
В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.
Сайт курса: `https://compscicenter.ru/courses/algorithms-2/2015-spring/`
Семинары ведёт Сергей Владимирович Копелиович,
контакты: burunduk30@gmail.com, vk.com/burunduk1

В каждом условии указан таймлимит для `C/C++`. Таймлиминт для `Java` примерно в 2 раза больше. Таймлиминт для `Python` примерно в 4 раза больше.

06.Base [1/1]

Задача 6A. Суффиксный массив [0.35 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** (читайте “случайная строка”). 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

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

06.Advanced [2/4]

Задача 6B. LCP для суффиксного массива [0.4 sec, 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 ($1 \leqslant N \leqslant 10^5$). 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

sufflcp.in	sufflcp.out
5 cacao 2 4 1 3 5	1 0 2 0

Замечание

Суффиксный массив для строки cacao:

```
acao
ao
cacao
cao
o
```

Подсказка по решению

Можно посчитать хешами, можно алгоритмом Касаи за $\mathcal{O}(n)$.

Задача 6C. Циклические сдвиги [0.5 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 ($1 \leqslant k \leqslant 100\,000$).

Формат выходных данных

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

Пример

shifts.in	shifts.out
abacabac 4	cabacaba
abacabac 5	IMPOSSIBLE

Замечание

Кроме решений суффиксным массивом или деревом за $\mathcal{O}(n \log n)$, можно использовать хеши и k-ую порядковую статистику за линейное время.

Задача 6D. Подстроки [1 sec, 256 mb]

Дана строка s. Вам требуется подсчитать количество её различных подстрок. Пустую строку учитывать не следует.

Формат входных данных

В единственной строке входного файла содержится данная строка s, состоящая из строчных латинских букв. Длина строки не превосходит 20 000 символов.

Формат выходных данных

В единственной строке выходного файла выведите единственное число — количество различных подстрок s.

substr.in	substr.out
aaaa	4
abacaba	21

Замечание

Предполагается решение суффиксными массивом за $\mathcal{O}(n \log^2 n)$ *или аккуратно написанным сжатым суффиксными деревом за* $\mathcal{O}(n^2)$.

Задача 6E. Подстроки-3 [0.5 sec, 256 mb]

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

Формат входных данных

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

Формат выходных данных

Наибольшая общая подстрока.

Примеры

substr3.in	substr3.out
3 abacaba mycabarchive acabistrue	cab

Замечание

Предполагается решение за $\mathcal{O}(n \log n)$: бинарный поиск + хеши + хеш-таблица.

06.Hard [0/2]

Задача 6F. Башни [0.2 sec, 256 mb]

Задано число n и последовательность из n чисел. Требуется рассмотреть все возможные циклические сдвиги заданной последовательности, отсортировать их в лексикографическом порядке, и вывести сумму наибольших общих префиксов соседних в этом порядке сдвигов.

Формат входных данных

Первая строка содержит целое число $1 \leqslant n \leqslant 50\,000$ — количество магических башен. Вторая строка содержит n чисел в интервале от 0 до 100 — заданную последовательность.

Формат выходных данных

Выведите одно число — искомую сумму.

Пример

towers.in	towers.out
11 12 8 18 18 8 18 18 8 15 15 8	13

Замечание

Предполагается решение за $\mathcal{O}(n \log n)$ суффиксным массивом. Решение за $\mathcal{O}(n \log^2 n)$ при хорошем написании тоже может зайти.

Задача 6G. Ненокку [1.2 sec, 256 mb]

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово "книга". Но он не любит читать книги (он лучше полазает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

Формат входных данных

В каждой строчке входного файла записано одна из двух записей.

1. `? <слово>` (<слово> — это набор не более 50 латинских символов);
2. `A <текст>` (<текст> — это набор не более 10^5 латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведение.
2 означает добавление в произведение <текст>.
Писатель только начал работать над произведением, поэтому он не мог написать более 10^5 символов. Суммарная длина всех запросов не превосходит 15 мегабайт плюс 12140 байт.

Формат выходных данных

Выведите на каждую строчку типа 1 "`YES`", если существует подстрока <слово>, и "`NO`" в противном случае. Не следует различать регистр букв.

Пример

nenokku.in	nenokku.out
? love ? is A Loveis ? love ? WHO A Whoareyou ? is	NO NO YES NO YES

Замечание

Задачу можно сдать или Укконеном, или суффиксным автоматом, или аккуратно написаными хешами.