

Задача А. Суффиксный бор (1 балл)

Имя входного файла: `trie.in`
Имя выходного файла: `trie.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Постройте суффиксный бор данной строки s .

Формат входного файла

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100$). Строка состоит из строчных латинских букв.

Формат выходного файла

В первой строке выходного файла выведите два натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном боре соответственно. В следующих m строках выведите описания ребер в формате `<родитель> <потомок> <символ>`. Корнем бора должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n .

Примеры

<code>trie.in</code>	<code>trie.out</code>
ababb	12 11 1 2 a 1 7 b 2 3 b 3 4 a 3 11 b 4 5 b 5 6 b 7 8 a 7 12 b 8 9 b 9 10 b

Задача В. Суффиксный массив (2 балла)

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

Постройте суффиксный массив для заданной строки s .

Формат входного файла

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

Формат выходного файла

Выведите $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

Примеры

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4

Задача С. Суффиксное дерево (2 балла)

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

Постройте суффиксное дерево для заданной строки s .

Формат входного файла

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$). Строка состоит из строчных латинских букв.

Формат выходного файла

В первой строке выходного файла выведите два натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном дереве соответственно. В следующих m строках выведите описания ребер в формате $\langle \text{родитель} \rangle \langle \text{потомок} \rangle \langle l \rangle \langle r \rangle$. Эта запись означает, что на ребре написана строка $s[l..r]$, при этом значение l должно быть минимально возможным. Корнем дерева должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n .

Примеры

<code>tree.in</code>	<code>tree.out</code>
ababb	7 6 1 4 1 2 1 6 2 2 4 2 3 5 4 5 5 5 6 3 3 5 6 7 5 5

Задача D. Суффиксный автомат (бонус, 2 балла)

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

Чтобы найти информацию по суффиксным автоматам в интернете на английском языке, рекомендуем использовать аббревиатуру DAWG (directed acyclic word graph).

Постройте суффиксный автомат для заданной строки s .

Формат входного файла

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$). Строка состоит из строчных латинских букв.

Формат выходного файла

В первой строке выходного файла выведите три натуральных числа n и m , разделенных пробелом — число вершин и ребер в суффиксном автомате соответственно. В следующих m строках выведите описания ребер в формате `<родитель> <потомок> <символ>`. Начальным состоянием автомата должна быть вершина с номером 1. Вершины должны быть занумерованы натуральными числами, не превышающими n . Затем выведите f — количество допускающих состояний автомата. На следующей строке выведите номера допускающих состояний автомата.

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. Рефрен

Имя входного файла: `refrain.in`
Имя выходного файла: `refrain.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

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 входного файла

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 выходного файла

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

<code>refrain.in</code>	<code>refrain.out</code>
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Задача F. Количество подстрок (2 балла)

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

Вычислите количество различных подстрок строки s .

Формат входного файла

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

Формат выходного файла

Выведите одно число — ответ на задачу.

Примеры

<code>count.in</code>	<code>count.out</code>
ababb	11

Задача G. Наибольшая общая подстрока (2 балла)

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

Найдите наибольшую общую подстроку строк s и t .

Формат входного файла

Первая строка входного файла содержит строку s , вторая — t ($1 \leq |s|, |t| \leq 100,000$). Строки состоят из строчных латинских букв.

Формат выходного файла

Выведите одну строку — наибольшую общую подстроку строк s и t . В случае, если ответ не единственный, выведите минимальный лексикографически.

Примеры

<code>common.in</code>	<code>common.out</code>
ababb abacabba	aba