

Содержание

Обязательные задачи	3
Задача 27А. Обратное по модулю [0.2 sec, 256 mb]	3
Задача 27В. Ретроанализ для маленьких [1.0 sec, 256 mb]	4
Задача 27С. Жестокая задача [0.2 sec, 256 mb]	5
Задача 27D. Суффиксный массив [0.2 sec, 256 mb]	6
Задача 27Е. LCP для суффиксного массива [0.5 sec, 256 mb]	7
Задача 27F. Циклические сдвиги [0.5 sec, 256 mb]	8
Задача 27G. Общая подстрока [2.0 sec, 256 mb]	9
Задача 27Н. Суффиксный массив (3 балла) [1.0 sec, 256 mb]	10
Бонус	11
Задача 27I. Суффиксное дерево [1.0 sec, 256 mb]	11
Задача 27J. Ненокку [2.0 sec, 256 mb]	12
Задача 27K. Рефрен [1.5 sec, 256 mb]	13
Задача 27L. Суффиксный пулемёт [0.2 sec, 256 mb]	14
Задача 27M. Монетки [0.2 sec, 256 mb]	16

Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html

http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html

Выделение памяти.

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

Обязательные задачи

Задача 27А. Обратное по модулю [0.2 сек, 256 mb]

Даны два целых числа — a, m ($0 \leq a < m$).

Нужно найти такое целое x , что $ax \equiv 1 \pmod m$

Формат входных данных

На первой строке два целых числа — a, m ($0 \leq a < m \leq 10^{18}$).

Формат выходных данных

Если такого x не существует, выведите -1 . Иначе выведите целое x ($0 \leq x < m$). Если ответов несколько, выведите любой.

Примеры

inverse.in	inverse.out
7 30	13

Замечание

Типа `long long` Вам должно хватить.

Задача 27В. Ретроанализ для маленьких [1.0 sec, 256 mb]

Дан ориентированный весёлый граф из n вершин и m ребер. Оля и Коля в игру. Изначально фишка стоит в вершине i . За ход можно передвинуть фишку по любому из исходящих ребер. Тот, кто не может сделать ход, проигрывает. Ваша задача — для каждой вершины i определить, кто выиграет при оптимальной игре обоих.

Формат входных данных

Входные данные состоят из одного или нескольких тестов. Каждый тест содержит описание весёлого ориентированного графа. Граф описывается так: на первой два целых числа n ($1 \leq n \leq 300\,000$) и m ($1 \leq m \leq 300\,000$). Следующие m строк содержат ребра графа, каждое описывается парой целых чисел от 1 до n . Пара $a\ b$ обозначает, что ребро ведет из вершины a в вершину b . В графе могут быть петли, могут быть кратные ребра. Сумма n по всем тестам не превосходит 300 000, сумма m по всем тестам также не превосходит 300 000.

Формат выходных данных

Для каждого теста выведите для каждой вершины **FIRST**, **SECOND** или **DRAW** в зависимости от того, кто выиграет при оптимальной игре из этой вершины. Ответы к тестам разделяйте пустой строкой.

Примеры

retro.in	retro.out
5 5	DRAW
1 2	DRAW
2 3	DRAW
3 1	FIRST
1 4	SECOND
4 5	
2 1	FIRST
1 2	SECOND
4 4	
1 2	FIRST
2 3	FIRST
3 1	SECOND
1 4	SECOND

Замечание

Помните, как писать bfs? Кстати, можно и dfs...

Задача 27С. Жестокая задача [0.2 sec, 256 mb]

Штирлиц и Мюллер стреляют по очереди. В очереди n человек, стоящих друг за другом. Каждым выстрелом убивается один из стоящих. Кроме того, если у кого-то из стоящих в очереди убиты все его соседи, то этот человек в ужасе убегает. Проигрывает тот, кто не может ходить. Первым стреляет Штирлиц. Требуется определить, кто выиграет при оптимальной игре обеих сторон, и если победителем будет Штирлиц, то найти все возможные первые ходы, ведущие к его победе.

Формат входных данных

Входной файл содержит единственное число n ($2 \leq n \leq 5000$) — количество человек в очереди.

Формат выходных данных

Если выигрывает Мюллер, выходной файл должен состоять из единственного слова `Mueller`. Иначе в первой строке необходимо вывести слово `Schtirlitz`, а в последующих строках — номера людей в очереди, которых мог бы первым ходом убить Штирлиц для достижения своей победы. Номера необходимо выводить в порядке возрастания.

Пример

<code>cruel.in</code>	<code>cruel.out</code>
3	Schtirlitz 2
4	Mueller
5	Schtirlitz 1 3 5

Замечание

Если Вы были на паре практики, задача должна быть для Вас очевидной.

Задача 27D. Суффиксный массив [0.2 сек, 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

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

Замечание

Это *очень* простая задача. Никаких сложных суффструктур писать не нужно.

Задача 27Е. LCP для суффиксного массива [0.5 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

sufflcp.in	sufflcp.out
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.

Касаи пишет короче, чем хеши.

Задача 27F. 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 ($1 \leq k \leq 100\,000$).

Формат выходных данных

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

Пример

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

Замечание

Хеши можно сдать. Но только за $\mathcal{O}(n \log n)$.

Задача 27G. Общая подстрока [2.0 сек, 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

common.in	common.out
0100 0010 3	01

Замечание

Обычного несжатого дерева вполне хватит.

Задача 27Н. Суффиксный массив (3 балла) [1.0 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 $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

Примеры

array.in	array.out
ababb	1 3 5 2 4

Замечание

Ничего лишнего. $\mathcal{O}(n \log n)$.

Бонус

Задача 271. Суффиксное дерево [1.0 sec, 256 mb]

Дана строка s . Постройте сжатое суффиксное дерево для строки s и выведите его. Найдите такое дерево, которое содержит минимальное количество вершин.

Формат входных данных

В первой строке записана строка s ($1 \leq |s| \leq 10^5$), последний символ строки доллар «\$», остальные символы строки маленькие латинские буквы.

Формат выходных данных

Пронумеруйте вершины дерева от 0 до $n - 1$ в порядке обхода в глубину, обходя поддеревья в порядке лексикографической сортировки исходящих из вершины рёбер. Используйте ASCII-коды символов для определения их порядка.

В первой строке выведите число n – количество вершин дерева. В следующих $n - 1$ строках выведите описание вершин дерева, кроме корня, в порядке увеличения их номеров.

Описание вершины дерева v состоит из трёх целых чисел: p, lf, rf , где p ($0 \leq p \leq n, p \neq v$) – номер родителя текущей вершины. На ребер ведущем из p в v написана подстрока $s[lf..rf)$ ($0 \leq lf < rf \leq |s|$).

Примеры

suftree.in	suftree.out
aaa\$	7 0 3 4 0 0 1 2 3 4 2 1 2 4 3 4 4 2 4
b\$	3 0 1 2 0 0 2
ababa\$	10 0 5 6 0 0 1 2 5 6 2 1 3 4 5 6 4 3 6 0 1 3 7 5 6 7 3 6

Замечание

А кто сдаст автоматом или массивом – молодец ;-)

Задача 27J. Ненокку [2.0 сек, 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 ползает 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	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

Замечание

Можно затолкать хеши. Но вообще предполагается Укконен или автомат.

Задача 27К. 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

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 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

refrain.in	refrain.out
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

Замечание

Та самая задача, когда дерево проще массива.

Задача 27L. Суфффиксный пулемёт [0.2 sec, 256 mb]

Или зачёт, или автомат.

Ганнибал Ректор

Теоретическая подготовка новобранцев армии Поссилтума включала в себя не только занятия по военному праву, но и начала криптографии. Лекции читал майор Мега Байт, не чуждый солдатского юмора. Гвидо и Нунцио, в чьё задание входил развал армии Поссилтума изнутри, решили на этом сыграть, внеся путаницу в терминологию. В начале очередной лекции Нунцио поднял руку и спросил:

— Вот вы на прошлой лекции рассказывали про конечные автоматы. А про конечные пулемёты расскажете?

Мега Байт не растерялся.

— Суфффиксный пулемёт — это конечный автомат, принимающий все суфффиксы данной строки (от нулевого до L -го включительно, где L — длина строки), и только их. Сержант Гвидо!

— Я, господин майор!

— Вы сможете отличить автомат от пулемёта?

— Так точно, господин майор!

— Вам дан конечный автомат. Требуется проверить, является ли он суфффиксным пулемётом данной строки.

К сожалению, написание программ такого типа не входило в обязанности Гвидо и Нунцио как в Синдикате, так и в корпорации М. И. Ф. Так что соответствующую программу придётся писать Вам.

Формат входных данных

Во входном файле задан один или несколько тестовых наборов. В первой строке каждого набора заданы количество состояний автомата N , количество переходов M , а также количество принимающих состояний T ($1 \leq T \leq N \leq 50\,000$, $1 \leq M \leq 100\,000$). Во второй строке через пробел заданы T различных чисел в пределах от 1 до N — принимающие состояния автомата, в возрастающем порядке. В последующих M строках заданы переходы в виде $a_i b_i c_i$, где $1 \leq a_i, b_i \leq n$, а c_i — маленькая буква латинского алфавита. Переход производится из состояния a_i в состояние b_i по букве c_i . Из каждого состояния a_i есть не более одного перехода по символу c_i . Последняя строка описания набора — это строка S , для которой автомат должен являться пулемётом. Она состоит только из маленьких латинских букв, и её длина лежит в пределах от 1 до 50 000 включительно. Кроме того, сумма всех N и суммарная длина всех строк, для которых необходимо произвести проверку, не превосходит 50 000, а сумма всех M не превосходит 100 000.

Файл заканчивается фиктивным набором, в котором $N = M = T = 0$.

Начальным состоянием автомата является первое. Если при интерпретации какой-то строки в автомате отсутствует соответствующий переход, то автомат вываливается по ошибке и строку не принимает. Таким образом, строка принимается, только если при её интерпретации были найдены все переходы, и по их завершении автомат оказался в принимающем состоянии (при этом неважно, были по пути принимающие состояния, или нет).

Формат выходных данных

Выведите в выходной файл, является ли данный автомат пулемётом, следуя формату примера.

Пример

suffix.in
2 1 2 1 2 1 2 a a 2 2 2 1 2 1 1 a 1 2 b ab 0 0 0
suffix.out
Automaton 1 is a machinegun. Automaton 2 is not a machinegun.

Замечание

Для решения задачи знание и понимание суффавтомата не обязательно.

Задача 27M. Монетки [0.2 sec, 256 mb]

На столе лежат в ряд N кучек монеток. В i -й кучке лежит ровно a_i монеток. При этом оказалось, что $a_i \leq a_j$ при $i < j$.

Катя и Серёжа играют с этими монетками в игру. За ход можно взять любое ненулевое количество монеток из любой кучки, но условие ($a_i \leq a_j$ при $i < j$) должно сохраниться. Выигрывает, как обычно, взявший последнюю монету.

Вам требуется определить, кто выиграет при правильной игре. Серёжа ходит первый.

Формат входных данных

В первой строке задано число N ($1 \leq N \leq 100\,000$). В следующей строке задано N чисел — a_1, a_2, \dots, a_N ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Если выигрывает Серёжа, выведите «Sergey», иначе выведите «Katya».

Пример

coins.in	coins.out
1 10	Sergey
2 10 15	Sergey
2 10 10	Katya