

Задача А. Размеры поддеревьев

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам дано дерево в корне с вершиной 1.

Поддеревом вершины v называется такое наибольшее множество вершин, в котором путь от каждой вершины до корня содержит вершину v .

Для каждой вершины выведете размер ее поддерева.

Формат входных данных

В первой строке вам дано число n - размер дерева ($2 \leq n \leq 10^6$).

Затем в следующей строке следует описание массива p размера $n - 1$. В нем p_i (где $1 \leq i \leq n - 1$) означает, что в дереве есть ребро между вершинами $i + 1$ и p_i ($1 \leq p_i \leq i$).

Формат выходных данных

Для каждой вершины выведете размер ее поддерева.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| 6 1 1 2 2 3 | 6 3 2 1 1 1 |

Задача В. Кратчайший путь

Имя входного файла: dag-shortpath.in
Имя выходного файла: dag-shortpath.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан ориентированный взвешенный ациклический граф. Требуется найти в нем кратчайший путь из вершины s в вершину t .

Формат входных данных

Первая строка входного файла содержит четыре целых числа n , m , s и t — количество вершин, дуг графа, начальная и конечная вершина соответственно. Следующие m строк содержат описания дуг по одной на строке. Ребро номер i описывается тремя целыми числами b_i , e_i и w_i — началом, концом и длиной дуги соответственно ($1 \leq b_i, e_i \leq n$, $|w_i| \leq 1000$).

Входной граф не содержит циклов и петель.

$1 \leq n \leq 100\,000$, $0 \leq m \leq 200\,000$, $1 \leq s, t \leq n$.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — длину кратчайшего пути из s в t . Если пути из s в t не существует, выведите `Unreachable`.

Примеры

| dag-shortpath.in | dag-shortpath.out |
|--------------------|-------------------|
| 2 1 1 2 1 2 -10 | -10 |
| 2 1 2 1 1 2 -10 | Unreachable |

Задача С. Невозрастающая подпоследовательность

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

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длиной $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число k — длину максимальной невозрастающей подпоследовательности. Во второй строке должны быть выведены все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Примеры

| <code>subseq.in</code> | <code>subseq.out</code> |
|------------------------|-------------------------|
| 5 | 3 |
| 5 8 10 4 1 | 1 4 5 |

Задача D. Распил брусьев

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Вам нужно распилить деревянный брус на несколько кусков в заданных местах. Распилочная компания берет k рублей за распил одного бруска длиной k метров на две части.

Понятно, что различные способы распила приводят к различной суммарной стоимости заказа. Например, рассмотрим брус длиной 10 метров, который нужно распилить на расстояниях 2, 4 и 7 м, считая от одного конца. Это можно сделать несколькими способами. Можно распилить сначала на отметке 2 м, потом 4 и, наконец, 7 м. Это приведет к стоимости $10 + 8 + 6 = 24$, потому что сначала длина бруса, который пилили, была 10 м, затем она стала 8 м, и, наконец, 6 м. А можно распилить иначе: сначала на отметке 4 м, затем 2, затем 7 м. Это приведет к стоимости $10 + 4 + 6 = 20$, что лучше.

Определите минимальную стоимость распила бруса на заданные части.

Формат входных данных

Первая строка входных данных содержит целое число L ($2 \leq L \leq 10^6$) - длину бруса и целое число N ($1 \leq N \leq 100$) - количество распилов. Во второй строке записано N целых чисел C_i ($0 < C_i < L$) в строго возрастающем порядке - места, в которых нужно сделать распилы.

Формат выходных данных

Выведите одно натуральное число - минимальную стоимость распила.

Примеры

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 10 3 2 4 7 | 20 |
| 100 3 15 50 75 | 200 |

Задача Е. Шоколадка

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

Команда «Отбой» участвует в очередном марафоне по «Угадай мелодию. Rock version». Чтобы было чем подкрепиться во время игры, команда взяла с собой большую прямоугольную плитку шоколада размерами $w \times h$. У команды есть список из n пар чисел — размеры шоколадок, которые команда считает счастливыми. Прежде чем приступить к поеданию шоколадки, участники команды решили поделить имеющуюся плитку на счастливые шоколадки. Для этого они действуют следующим образом: сначала плитка шоколада ломается на 2 части по линии, строго параллельной одной из своих сторон, после чего каждую из полученных частей они могут продолжить ломать аналогичным образом.

Вам поручили определить, какое максимальное количество счастливых шоколадок команда сможет получить, действуя по данной схеме. Шоколадки, полученные поворотом счастливых, счастливыми не являются.

Формат входных данных

В первой строке входного файла заданы три целых числа w, h, n — размеры плитки шоколада и количество вариантов размера счастливых шоколадок соответственно ($1 \leq w, h \leq 300, 1 \leq n \leq w \times h$). В следующих n строках заданы пары целых чисел w_i, h_i — размеры счастливых шоколадок ($1 \leq w_i \leq w, 1 \leq h_i \leq h$).

Формат выходных данных

В единственную строку выходного файла выведите максимальное количество счастливых шоколадок, на которые можно разрезать данную плитку.

Примеры

| <code>chocolate.in</code> | <code>chocolate.out</code> |
|--|----------------------------|
| 21 11 4 10 4 6 2 7 5 15 10 | 15 |
| 9 12 5 1 12 2 6 3 4 4 3 6 2 | 9 |

Задача F. Пастбища

Имя входного файла: `pasture.in`
Имя выходного файла: `pasture.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Фермер Джон решил снабдить каждую из своих коров сотовым телефоном. Для этого ему требуется установить сотовые станции на его N пастбищах, последовательно пронумерованных от 1 до N .

Ровно $N - 1$ пара пастбищ являются соседними, и для любых двух пастбищ A и B имеется последовательность соседних пастбищ таких, что A — первое пастбище этой последовательности, а B — последнее. Сотовые станции размещаются только на пастбищах и имеют достаточный радиус действия, чтобы обеспечить связью это пастбище и все соседние.

Помогите фермеру Джону определить, какое минимальное количество станций он должен установить, чтобы обеспечить связью все пастбища.

Формат входных данных

На первой строке входного файла находится одно целое число N ($1 \leq N \leq 100\,000$). Далее следуют $N - 1$ строк, каждая из которых содержит пару целых чисел — очередную пару соседних пастбищ A и B ($1 \leq A \leq N$; $1 \leq B \leq N$; $A \neq B$).

Формат выходных данных

Выведите в выходной файл одно число — минимальное достаточное количество станций.

Примеры

| <code>pasture.in</code> | <code>pasture.out</code> |
|-------------------------|--------------------------|
| 5 | 2 |
| 1 3 | |
| 5 2 | |
| 4 3 | |
| 3 5 | |

Задача G. Контест-палиндром

| | |
|-------------------------|-------------------|
| Имя входного файла: | стандартный ввод |
| Имя выходного файла: | стандартный вывод |
| Ограничение по времени: | 3 секунды |
| Ограничение по памяти: | 256 мегабайт |

Как вы знаете, Хет и Юра очень любят готовить контесты и готовы заниматься этим целыми часами. Вот и сейчас Хет перебирает задачи, которые он хочет добавить в предстоящий контест. Он считает, что контест будет **успешным** только в том случае, если первые буквы названий задач, которые в него входят, образуют палиндром (контест, который не содержит ни одной задачи, по логике Хета, тоже успешный!).

Система подготовки контестов «Моногон» содержит в себе N задач, которые пронумерованы последовательно от 1 до N . Все их названия начинаются с заглавной латинской буквы. Хет хочет выбрать некоторые из них. При этом, «Моногон» требует, чтобы задачи в контесты были отсортированы по порядковому номеру. Т. е. чтобы номера задач в контесте образовывали возрастающую последовательность.

Хет выписал первые буквы названий задач с номера 1 по N и принёс вам. Помогите Хету определить, сколько всего **успешных** контестов он сможет подготовить.

Так как количество контестов может быть очень большим, ответ требуется вывести по модулю $10^9 + 7$.

Формат входных данных

Единственная строка представляет из себя последовательность длины N ($1 \leq N \leq 5000$) из латинских заглавных букв.

Формат выходных данных

Выведите кол-во успешных контестов по модулю $10^9 + 7$.

Примеры

| стандартный ввод | стандартный вывод |
|------------------|-------------------|
| АВАСАВА | 42 |
| ІТМО | 5 |

Замечание

В первом примере название задачи №1 начинается с буквы «А», №2 с буквы «В», №3 с буквы «А» и т. д.

Задача Н. Дубы

| | |
|-------------------------|-----------------------|
| Имя входного файла: | <code>oaks.in</code> |
| Имя выходного файла: | <code>oaks.out</code> |
| Ограничение по времени: | 1.5 секунды |
| Ограничение по памяти: | 256 мегабайт |

На аллее перед зданием Министерства Обороны в ряд высажены n дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубки, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубки аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

Формат входных данных

Первая строка входного файла содержит целое число n — количество дубов, растущих на аллее ($2 \leq n \leq 200$). Вторая строка содержит n чисел — высоты дубов, приведённые слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

Формат выходных данных

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число -1 .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число m — минимальное количество дубов, которые необходимо срубить. В следующие m строк выведите оптимальный план вырубки деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до n .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

Примеры

| <code>oaks.in</code> | <code>oaks.out</code> |
|----------------------|-----------------------|
| 5 | 2 |
| 3 2 4 8 5 | 2 |
| | 4 |