

Задача А. Декомпозиция

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.6 секунд
Ограничение по памяти: 256 мегабайт

Рассмотрим дерево T . Назовем деревом декомпозиции корневое дерево $D(T)$. Выберем любую из вершин дерева T , назовем ее r . Рассмотрим все компоненты связности дерева T , после удаления вершины r : S_1, S_2, \dots, S_k . Тогда корнем $D(T)$ будет вершина r , а детьми r в $D(T)$ будут $D(S_1), D(S_2), \dots, D(S_k)$.

Вам дано дерево T . Найдите дерево декомпозиции высоты не более 20. Высота дерева — максимальное число вершин в пути от корня до какой-то вершины.

Формат входных данных

Первая строка содержит n ($1 \leq 2 \cdot 10^5$) — количество вершин дерева.

Следующие $n - 1$ строк содержат пары чисел u_i, v_i ($1 \leq u_i, v_i \leq n$), описывающие рёбра дерева.

Формат выходных данных

Выведите n чисел, где i -е — родитель вершины i в дереве декомпозиции. Если вершина — корень, выведите 0.

Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3	2 0 2
9 3 2 4 2 1 2 5 1 1 6 7 6 6 8 8 9	0 1 2 2 1 1 6 6 8

Задача В. Красим дерево

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.6 секунд
Ограничение по памяти: 256 мегабайт

Дано взвешенное дерево. Вам необходимо выполнять 2 типа запросов:

- «1 $v d c$ » — покрасить все вершины на расстоянии не более d от v в цвет c . Изначально все вершины имеют цвет 0.
- «2 v » — вывести цвет вершины v .

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10^5$) — количество вершин в дереве.

Следующие $n - 1$ содержат тройки чисел u_i, v_i, w_i ($1 \leq u_i, v_i \leq n, 1 \leq w_i \leq 10^4$). i -е ребро соединяет вершины u_i, v_i и имеет вес w_i .

В следующей строке содержится количество запросов q ($1 \leq q \leq 10^5$).

Каждая из следующих q строк содержит запрос какого-то типа:

- 1 $v d c$ ($1 \leq v \leq n, 0 \leq d \leq 10^9, 0 \leq c \leq 10^9$).
- 2 v ($1 \leq v \leq n$).

Формат выходных данных

Для каждого запроса второго типа выведите ответ на него.

Примеры

стандартный ввод	стандартный вывод
2	20
1 2 1	10
4	
1 1 1 10	
1 1 0 20	
2 1	
2 2	
5	6
1 2 30	6
1 3 50	0
3 4 70	5
3 5 60	7
8	
1 3 72 6	
2 5	
1 4 60 5	
2 3	
2 2	
1 2 144 7	
2 4	
2 5	

Задача С. В бухгалтерии опять всё перепутали

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Лупа и Пупа пошли получать зарплату. Но в бухгалтерии опять всё перепутали. Лупа получил зарплату за Пупу, а Пупа . . .

Пупа не хочет получать за Лупу и хочет доказать бухгалтерии, что она не права.

Пупа работает в крупной компании «MST Inc.», занимающейся информационным сопровождением «Всеберляндской олимпиады школьников по информатике». В компании «MST Inc.» работает n сотрудников, причём у каждого из них, кроме самой «MST», есть ровно один непосредственный начальник и несколько (возможно ноль) непосредственных подчинённых.

Всеми начальниками сотрудника компании «MST Inc.» называется множество, состоящее из его непосредственного начальника и множества начальников его непосредственного начальника. Известно, что у каждого сотрудника кроме самой «MST», «MST» входит в множество начальников этого сотрудника.

Множеством подчинённых у сотрудника называется множество, состоящее из него самого и множеств подчинённых у всех непосредственных подчинённых данного сотрудника. В частности, все сотрудники входят в множество подчинённых у «MST».

Каждый месяц каждому сотруднику начисляется зарплата, причём немаленькая, ведь иначе ни один сотрудник не согласился бы работать с «MST». Известно, что в нулевой месяц работы организации, каждому сотруднику заплатили по c_i бурлей. В качестве поощрения сотрудников «MST» придумала следующее правило: В каждый из следующих m месяцев берётся сотрудник с номером a_i и берётся число s_i — сумма зарплат всех сотрудников во множестве его начальников и подчинённых (включая его самого). Если это число оказывалось слишком большим, s_i берётся по модулю $10^9 + 7$. После этого берётся сотрудник с номером b_i , и к зарплате всех сотрудников, входящих во множество его начальников и подчинённых (включая его самого) прибавляется число s_i . С учётом этого изменения платится зарплата в i -й месяц и пересчитывается зарплата в следующие месяцы.

Вернёмся к Пупе. Пупа хочет показать бухгалтерии компании «MST Inc.» что она всё перепутала, а для этого ему надо узнать, сколько же ему должны были заплатить в каждый из месяцев с нулевого по m -й. К сожалению, в гениальной системе поощрения, разработанной «MST», не может разобраться никто. Поэтому эту задачу поручили вам.

Формат входных данных

В первой строке входных данных даны 2 числа n и m ($1 \leq n, m \leq 10^5$) — число сотрудников компании «MST Inc.» и последний день, когда выплачивалась зарплата Пупе.

Во второй строке записано $n - 1$ число. i -е из них — номер непосредственного начальника сотрудника номер i (i принимает значения от 1 до $n - 1$). При этом «MST» имеет номер 0 и не имеет непосредственного начальника. Пупа имеет номер $n - 1$.

В третьей строке записано n чисел c_i ($1 \leq c_i \leq 10^9$) — зарплата i -го сотрудника в нулевой день.

В каждой из следующих m строк записано по 2 числа a_i и b_i ($0 \leq a_i, b_i \leq n - 1$) — номер человека, на основе которого происходит поощрение и номер человека, к подчинённым и начальникам которого поощрение применяется (более подробно описано в условии).

Формат выходных данных

В единственной строке выведите $m + 1$ число — зарплату Пупы в каждый из дней с 0-го по m -й. Напоминаем, что Пупа имеет номер $n - 1$. Обратите внимание, что зарплата **не считается** по модулю $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3 0 0 1 1 1 0 0 2 1 1 2	1 4 4 28
4 3 0 1 1 0 1 0 0 0 1 1 3 2 3	0 1 6 20

Замечание

Пояснение к первому примеру:

В первый день к зарплате каждого сотрудника прибавилось 3 бурля и зарплаты стали соответственно 4, 4, 4.

Во второй день к зарплате сотрудников с номерами 0, 1 прибавилось по 8 бурлей и зарплаты стали соответственно 12, 12, 4.

Во третий день к зарплате сотрудников с номерами 0, 2 прибавилось по 24 бурля и зарплаты стали соответственно 36, 12, 28.

Задача D. Найти количество

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.6 секунд
Ограничение по памяти: 256 мегабайт

Дано подвешенное дерево из n вершин с корнем в 1. Необходимо обработать q запросов (v_i, h_i) : найти количество вершин, лежащих в поддереве v_i , расстояние до которых от v_i равно h_i . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

Формат входных данных

Первая строка содержит n ($1 \leq n \leq 3 \cdot 10^5$).

Следующая строка содержит $n - 1$ число p_2, \dots, p_n ($1 \leq p_i \leq i$). p_i — отец вершины i .

Следующая строка содержит число q ($1 \leq q \leq 3 \cdot 10^5$).

Следующие q строк содержат числа v_i, h_i ($1 \leq v_i \leq n, 1 \leq h_i \leq n$).

Формат выходных данных

Для каждого запроса выведите ответ.

Пример

стандартный ввод	стандартный вывод
5	2
1 2 2 1	1
3	0
1 1	
2 0	
3 5	

Замечание

А за сколько работает решение? Пишите @qoo2p5, если хотите проверить себя. :)

Задача Е. Дерево

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Дано дерево из n вершин и q запросов.

Каждый запрос начинается с трех целых чисел k , m и r , и продолжается k вершинами дерева a_1, a_2, \dots, a_k . Чтобы ответить на запрос, предположите, что дерево подвешено за вершину r . Рассмотрим разбиения данных k вершин на **не более чем** m групп так, что выполняются следующие условия:

- Каждая вершина принадлежит ровно одной группе, каждая группа содержит хотя бы одну вершину.
- Ни в одной группе нет двух вершин таких, что одна является предком (не обязательно непосредственным) другой.

Выведите количество различных таких разбиений по модулю $10^9 + 7$ для каждого запроса.

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n, q \leq 10^5$) — количество вершин в дереве и количество запросов, соответственно.

Каждая из следующих $n - 1$ вершин содержит два целых числа u и v ($1 \leq u, v \leq n, u \neq v$), обозначающие ребро между вершинами u и v . Гарантируется, что данный граф является деревом.

Каждая из следующих q строк начинается с трех целых чисел k , m и r ($1 \leq k, r \leq n, 1 \leq m \leq \min(300, k)$) — количество вершин, максимальный размер группы и корень дерева для данного запроса, соответственно. После этого следуют k различных целых чисел a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$) — вершины текущего запроса.

Гарантируется, что сумма значений k по всем запросам не превосходит 10^5 .

Формат выходных данных

Выведите q строк, где i -я строка содержит ответ на i -й запрос.

Примеры

стандартный ввод	стандартный вывод
7 2 5 4 2 6 5 3 1 2 7 5 4 6 3 3 2 7 4 3 3 1 4 6 2 1	2 0
7 2 4 7 2 5 4 1 5 1 5 6 4 3 3 3 2 7 1 4 2 1 6 3 2	1 1
5 2 3 5 4 5 4 2 1 4 2 2 3 1 2 2 2 4 5 4	2 1

Замечание

Рассмотрим первый пример.

В первом запросе нужно разделить три данные вершины (7, 4 и 3) на не более чем три группы, считая, что корнем дерева является вершина 2. Когда дерево подвешено за вершину 2, вершина 4 является предком вершин 3 и 7. Поэтому нельзя все вершины отнести к одной группе. Есть только 1 способ разделить эти вершины на две группы: [4] и [3, 7]. Кроме того, есть один способ разделить данные вершины на три группы: [7], [4] и [3]. Таким образом, есть всего 2 способа разбить данные вершины на не более чем три группы.

Во втором запросе дерево подвешено за вершину 4, при этом 6 является предком 2, а 2 является предком 1. Поэтому нельзя все вершины отнести к одной группе.

Задача F. Джон и волшебная коробка

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

Это интерактивная задача.

У Джона есть массив из n загадочных чисел. Ещё у него есть волшебная коробка, которая может сделать из двух чисел одно (исходные числа остаются на месте). Пусть $x \circ y$ — результат работы коробки с числами x и y . После долгих экспериментов Джон обнаружил два свойства коробки:

- $x \circ y = y \circ x$
- $x \circ (y \circ z) = (x \circ y) \circ z$

У Джона есть q племянниц, и каждая из них любит все числа из массива Джона, кроме некоторых k из них. Джон хочет подарить каждой племяннице результат комбинации с помощью коробки всех чисел из его массива, кроме тех k , которые эта племянница не любит.

Джон любит своих племянниц, но из-за многочисленных экспериментов волшебная коробка изнасилась. Джон не сможет использовать её больше, чем $4(n + q + q \cdot k)$ раз. Помогите ему получить все требуемые комбинации!

Протокол взаимодействия

В начале работы решение получает строку с тремя целыми числами n , q и k ($1 \leq q \leq n \leq 2000$, $n \geq 3$, $2 \leq 2^k \leq n$). Далее следует строка, содержащая n целых чисел a_1, \dots, a_n ($0 \leq a_i < 2^{50}$). После этого вы можете выполнять три типа запросов (каждый запрос должен быть выведен на отдельной строке):

- «? x y », где x и y — целые числа ($0 \leq x, y < 2^{50}$). Запрос этого типа служит для использования волшебной коробки. Ответ на него — строка, содержащая число $x \circ y$ ($0 \leq x \circ y < 2^{50}$). Вы можете использовать запрос такого типа не более $4(n + q + q \cdot k)$ раз. **Если вы превысите этот лимит, вам будет выдано число «-1».**
- «next». Запрос такого типа позволяет узнать k элементов массива, которые не любит очередная племянница. Ответ на него — строка из k различных целых чисел d_1, \dots, d_k ($1 \leq d_i \leq n$): индексы элементов, которые не любит эта племянница. *Вы можете снова использовать запрос такого типа только после того, как ответите на предыдущий запрос этого типа.*
- «! x », где x — это целое число ($0 \leq x < 2^{50}$). Здесь x должно быть тем числом, которое Джон подарит последней описанной племяннице (то есть племяннице, соответствующей самому недавнему запросу “next”). На этот запрос интерактор не отвечает.

Чтобы предотвратить буферизацию, очищайте буфер после вывода каждой строки: это может быть сделано, например, с помощью `fflush (stdout)` в C или C++, `System.out.flush ()` в Java, `flush (output)` в Pascal или `sys.stdout.flush ()` в Python. А ещё не забывайте завершать каждую строку символом перевода строки.

Пример

стандартный ввод	стандартный вывод
3 3 1	
0 1 0	
1	next
1	? 1 0
2	! 1 next
0	? 0 0
3	! 0 next ! 1

Замечание

В каждом тесте устройство волшебной коробки фиксировано и не зависит от ваших запросов. В разных тестах используются разные устройства. Гарантируется, что волшебная коробка удовлетворяет свойствам, описанным в условии.

В примере операция, которую проделывает волшебная коробка — это операция побитового ИЛИ.

Задача G. Найти ближайшую

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано дерево из n вершин, цвет i -й вершины равен a_i . Необходимо обработать q запросов (v_i, c_i) : найти расстояние от v_i до ближайшей вершины цвета c_i . Расстояние между вершинами — минимальное количество рёбер в пути между ними.

Формат входных данных

Первая строка содержит n ($1 \leq n \leq 10^5$).

Следующая строка содержит $n - 1$ число p_1, \dots, p_{n-1} ($0 \leq p_i < i$). p_i — отец вершины i .

Следующая строка содержит числа a_1, \dots, a_n ($0 \leq a_i < n$).

Следующая строка содержит число q ($1 \leq q \leq 10^5$).

Следующие q строк содержат числа v_i, c_i ($0 \leq v_i < n, 0 \leq c_i < n$).

Формат выходных данных

Для каждого запроса выведите расстояние до ближайшей вершины требуемого цвета, или -1 , если такой нет.

Пример

стандартный ввод	стандартный вывод
5	0 1 2 -1 2 1 2 1 1
0 1 1 3	
1 2 3 2 1	
9	
0 1	
0 2	
0 3	
1 0	
2 1	
2 2	
3 3	
3 1	
4 2	

Задача Н. Гоша и праздники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Как известно, жители планеты Иннополис — очень педантичные люди. И даже когда дело касается праздников, они всегда хотят быть уверенными в том, что все пройдёт как по маслу. Так, расписание празднований всех событий на этой планете составлено почти на три миллиона лет вперёд! Гоша — большой любитель праздников. Он решил прилететь в какой-то из городов планеты Иннополис и посетить как можно больше праздников.

На планете Иннополис n городов, соединённых $n - 1$ двунаправленными дорогами так, что из любого города планеты можно добраться до любого другого, возможно, посещая другие города. Каждое событие на Иннополисе характеризуется номером города c_i , в котором оно будет отпраздновано, и номером дня d_i , в который его будут праздновать.

Гоша настолько везучий человек, что день его прибытия на планету имеет номер 0 в календаре планеты Иннополис, причём исходно он может прилететь в любой город планеты. Гоша решил узнать, какое максимальное количество праздников он может посетить на этой планете. Для этого он обратился за помощью к вам.

Формат входных данных

В первой строке входного файла задано одно число n ($n \geq 1$) — количество городов Иннополиса.

В следующих $n - 1$ строках заданы описания дорог, каждая дорога задается числами a_i , b_i и l_i ($1 \leq a_i, b_i \leq n$; $l_i \geq 1$) — номера городов, которые соединяет дорога и число дней, необходимых на ее преодоление.

В следующей строке задано число m ($m \geq 1$) — число праздников на планете.

В следующих m строках заданы пары чисел c_i и d_i ($1 \leq c_i \leq n$; $d_i \geq 1$) — номер города и номер дня, в который пройдёт i -й праздник.

Ограничения: $n \leq 2 \cdot 10^5$, $m \leq 2 \cdot 10^5$, $l_i \leq 10^9$, $d_i \leq 10^9$.

Формат выходных данных

В единственной строке выходного файла выведите одно число — максимальное количество праздников, которое может посетить Гоша.

Примеры

стандартный ввод	стандартный вывод
4 1 2 1 2 3 1 2 4 3 4 1 3 2 4 3 1 4 5	3
11 2 1 2 3 2 5 4 1 5 5 2 4 6 5 1 7 1 2 8 3 4 9 6 2 10 7 2 11 2 2 9 1 67 1 34 11 16 5 97 4 70 2 20 2 61 2 26 2 70	8
10 2 1 1 3 2 4 4 2 4 5 3 2 6 4 5 7 5 4 8 3 1 9 6 2 10 7 5 9 7 34 10 82 2 48 3 66 8 98 2 66 3 3 8 59 5 22	8

Задача I. Количество путей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Дано дерево из n вершин. Для каждого $d = 1 \dots n - 1$ найдите количество путей длины d .

Формат входных данных

Первая строка содержит n ($1 \leq n \leq 50000$) — количество вершин.

Следующие $n - 1$ строк содержат пары чисел u_i, v_i ($1 \leq u_i, v_i \leq n$), описывающие рёбра дерева.

Формат выходных данных

Выведите $n - 1$ число, где i -е — количество путей длины i .

Пример

стандартный ввод	стандартный вывод
3	2
1 2	1
2 3	

Задача J. Древландия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	512 мегабайт

В древландии есть города, и первый город — *столица*. Города соединены автобусными маршрутами так, что для города $i \neq 1$ есть автобус, который идёт оттуда в город p_i ($p_i < i$), именно в таком направлении.

В стране есть национальные блюда. Каждый город имеет своё специальное блюдо, оно и только оно может быть куплено там. Тип специального блюда для каждого города — один из типов национальных блюд.

Несколько друзей из нескольких городов хотят встретиться в одном городе для вечеринки. Они выбирают город такой, что, если они одновременно начнут туда идти, они встретятся там быстро, как только возможно. Путешествие на автобусе требует 1 единицу времени.

Они хотят купить некоторые блюда для вечеринки, соблюдая следующие требования:

1. Каждый друг должен купить одно и то же количество блюд.
2. Не должно быть двух блюд одного типа на вечеринке.
3. Каждый друг может купить только блюда, соответствующие городам, которые он посетил.

Для заданных запросов, найдите максимальное количество блюд, которое может быть на вечеринке.

Формат входных данных

Первая строка содержит три числа n , m , q .

- n — количество городов
- m — количество типов национальных групп
- q — количество запросов

Вторая строка содержит $n - 1$ чисел p_2, \dots, p_n , описывающие автобусные маршруты.

Третья строка содержит n чисел a_1, \dots, a_n , описывающие типы блюд, продающиеся в соответствующих городах.

Следующие строки содержат описания запросов. Каждый запрос описывается в следующем формате: число c , обозначающее количество друзей, а затем c чисел v_1, \dots, v_c . Пусть ответ на предыдущий ответ равен X (для первого запроса $X = 0$). Тогда друзья находятся в вершинах $(v_1 - 1 + X) \bmod n + 1, \dots, (v_c - 1 + X) \bmod n + 1$.

Не гарантируется, что для конкретного запроса все v_i различны.

- $2 \leq n \leq 3 \cdot 10^5$
- $1 \leq m \leq 1000$
- $1 \leq q \leq 5 \cdot 10^4$
- $1 \leq p_i < i$
- $1 \leq a_i \leq m$
- $2 \leq c \leq 5$
- $1 \leq v_i \leq n$

Формат выходных данных

Выведите q строк, i -я из которых — ответ на i -й запрос.

Примеры

стандартный ввод	стандартный вывод
5 3 4 1 2 2 1 2 3 1 3 1 2 3 4 3 5 2 2 4 3 4 2 5 2 2 2	2 0 0 0
11 6 3 1 2 2 4 5 4 5 8 9 4 5 6 1 1 2 3 2 3 4 5 2 3 3 10 8 4 6 5 10 10 2 9 6	6 4 2

Задача К. Peterson Polyglot

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Петрович обожает изучать новые языки, но самое любимое его увлечение — составление своих собственных. Языком Петрович называет множество слов, а словом — последовательность маленьких букв латинского алфавита.

Каждое утро Петрович составляет свой новый язык. Хранить все языки в явном виде очень сложно, поэтому Петрович придумал веник — специальную структуру данных для хранения языка, представляющую собой подвешенное дерево, на ребрах которого написаны буквы. Перед придумыванием языка веник представляет одну вершину — корень. При добавлении нового слова в язык Петрович встает в корень веника и обрабатывает буквы слова по одной. Пусть Петрович стоит в вершине u . Если из u есть ребро, на котором написана текущая буква, он переходит по нему. Иначе же, Петрович добавляет ребро из u в новую вершину v , пишет на нем текущую букву и переходит по этому ребру. Размером веника Петрович называет количество вершин в нем.

Вечером, приходя со смены, Петрович не может понять язык, придуманный утром: он ему кажется слишком сложным. Тогда Петрович старается упростить свой язык. Упрощением языка Петрович называет удаление букв из некоторых слов языка. Формально, Петрович фиксирует некоторое целое положительное число p , берет все слова, содержащие хотя бы p букв, и выкидывает из каждого из них букву с номером p . Буквы в слове Петрович предпочитает нумеровать, начиная с 1. Петрович считает, что при упрощении языка хотя бы одно слово должно измениться, то есть в языке должно быть хотя бы слово с длиной хотя бы p . Так как Петрович стремится сделать язык, придуманный утром, как можно проще, он старается подобрать число p таким образом, чтобы минимизировать размер веника, в котором он будет хранить язык.

Петровичу надоело заниматься одним и тем же каждый вечер, поэтому он обратился за помощью к вам. Напишите программу, которая будет находить минимальный размер веника, который может получиться в результате упрощения языка, придуманного Петровичем, и число p , которое нужно выбрать, чтобы получить такой размер.

Формат входных данных

В первой строке входного файла задано число n ($2 \leq n \leq 3 \cdot 10^5$) — размер веника языка Петровича.

В следующих $n - 1$ строках задано описание веника. В i -й из них записаны числа u_i, v_i и буква x_i , что соответствует ребру из u_i в v_i , на котором написана буква x_i .

Вершины пронумерованы числами от 1 до n . Все x_i являются маленькими буквами латинского алфавита. Вершина с номером 1 является корнем веника.

Гарантируется, что ребра описывают корректный веник, построенный по языку Петровича.

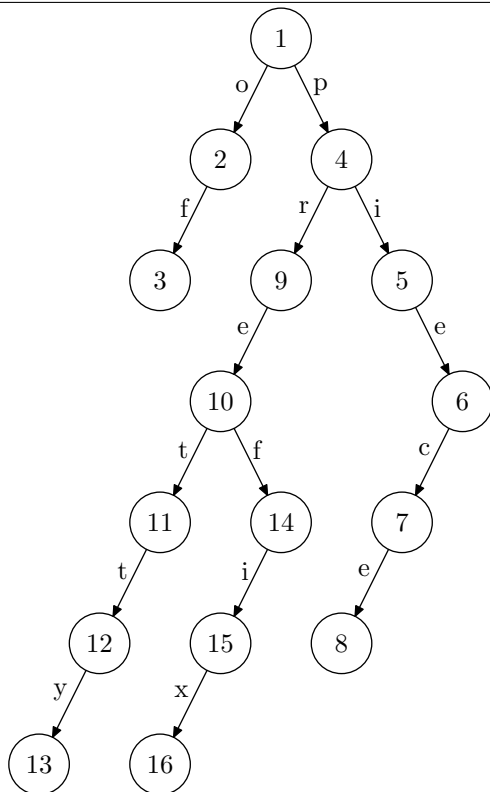
Формат выходных данных

В первой строке выведите минимальный возможный размер веника, который может получиться в результате упрощения языка.

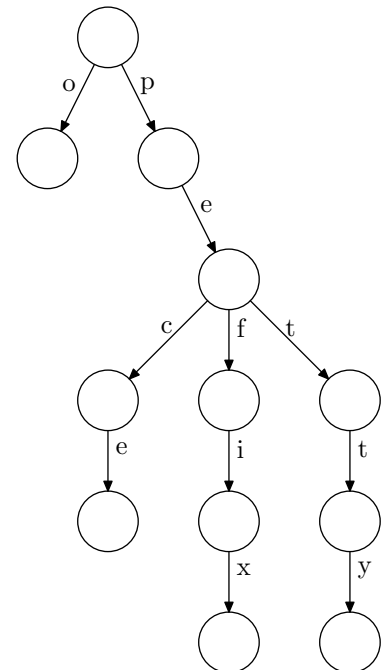
Во второй строке выведите число p , которое следует выбрать Петровичу для получения минимального размера. Если таких чисел p несколько, выведите минимальное из них.

Примеры

стандартный ввод	стандартный вывод
5 1 2 c 2 3 a 3 4 t 2 5 t	3 2
16 1 2 o 2 3 f 1 4 p 4 5 i 5 6 e 6 7 c 7 8 e 4 9 r 9 10 e 10 11 t 11 12 t 12 13 y 10 14 f 14 15 i 15 16 x	12 2



Веник для исходного языка из примера 2.



Веник после упрощения при $p = 2$.

Веник из второго примера может быть составлен из множества слов «piece», «of», «pie», «pretty», «prefix». После упрощения языка с $p = 2$ получается язык из слов «pесе», «о», «ре», «retty», «refix». Этот язык и задаёт веник минимального возможного размера.