

Задача А. Z-функция

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

Вам дана строка из строчных латинских букв. От вас требуется вычислить Z-функцию данной строки.

Формат входных данных

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает 10^6 .

Формат выходных данных

Требуется вывести Z-функцию данной строки.

Пример

стандартный ввод	стандартный вывод
abacaba	7 0 1 0 3 0 1

Замечание

Предполагается, что значение Z-функции для первого символа равно длине строки.

Задача В. Префикс-функция

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

Вам дана строка из строчных латинских букв. От вас требуется вычислить префикс-функцию данной строки.

Формат входных данных

Вводится строка, состоящая из строчных латинских букв. Длина строки не превышает 10^6 .

Формат выходных данных

Требуется вывести префикс-функцию данной строки.

Пример

стандартный ввод	стандартный вывод
abacaba	0 0 1 0 1 2 3

Замечание

Предполагается, что значение префикс-функции для первого символа равно нулю.

Задача С. Поиск подстроки (1 балл)

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

Найти все вхождения строки T в строку S .

Формат входных данных

Первые две строки входных данных содержат строки S и T , соответственно. Длины строк больше 0 и меньше 500 000, строки содержат только строчные латинские буквы.

Формат выходных данных

Выведите номера символов, начиная с которых строка T входит в строку S , в порядке возрастания.

Пример

стандартный ввод	стандартный вывод
ababbababa aba	0 5 7

Задача D. Неточное совпадение

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

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Пример

стандартный ввод	стандартный вывод
aaaa	4
Caabdaaaa	1 2 6 7

Задача Е. Мультимножество Василия

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

У автора уже закончились истории про Василия, поэтому он просто написал формальную постановку задачи.

У вас есть q запросов и мультимножество A , изначально содержащее только число 0. Запросы бывают трёх видов:

- «+ x » — добавить в мультимножество A число x .
- «- x » — удалить одно вхождение числа x из мультимножества A . Гарантируется, что хотя бы одно число x в этот момент присутствует в мультимножестве.
- «? x » — вам даётся число x , требуется вычислить $\max_{y \in A} x \oplus y$, то есть максимальное значение побитового исключающего ИЛИ (также известно как XOR) числа x и какого-нибудь числа y из мультимножества A .

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

Формат входных данных

В первой строке входных данных содержится число q ($1 \leq q \leq 200\,000$) — количество запросов, которые требуется обработать Василию.

Каждая из последующих q строк входных данных содержит один трёх символов «+», «-» или «?» и число x_i ($1 \leq x_i \leq 10^9$). Гарантируется, что во входных данных встречается хотя бы один запрос «?».

Обратите внимание, что число 0 всегда будет присутствовать в мультимножестве.

Формат выходных данных

На каждый запрос типа «?» выведите единственное целое число — максимальное значение побитового исключающего ИЛИ для числа x_i и какого-либо числа из мультимножества A .

Пример

стандартный ввод	стандартный вывод
10	11
+ 8	10
+ 9	14
+ 11	13
+ 6	
+ 1	
? 3	
- 8	
? 3	
? 8	
? 11	

Замечание

После первых пяти операций в мультимножестве A содержатся числа 0, 8, 9, 11, 6 и 1.

Ответом на шестой запрос будет число $11 = 3 \oplus 8$ максимальное из чисел $3 \oplus 0 = 3$, $3 \oplus 9 = 10$, $3 \oplus 11 = 8$, $3 \oplus 6 = 5$ и $3 \oplus 1 = 2$.

Задача F. Библиотека

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Прошел почти год с момента, как Рик оказался на Флорине, однако его сознание никак не прояснялось. Воспоминания о прошлом были спрятаны в глубинах его разума, а может и вовсе утеряны. Однако сегодня что-то случилось. Рик вспомнил: у него была работа. Он анализировал Ничто. Наверное, Ничто — это космос, а значит Рик в прошлом был космоаналитиком. А еще Рик вспомнил, что все жители Флорины должны были погибнуть, но он не знал, почему.

Резидента Мирлина Теренса заинтересовала эта информация, поэтому он взял Рика с собой в библиотеку Верхнего города. Может быть, какая-нибудь литература по космоанализу могла бы вернуть ему память? Теренс не знал, что пропавшего космоаналитика активно ищут, а потому в библиотеке был получен приказ сообщать о любых посетителях, которые спросят о такой литературе. Библиотекарь отследил запросы наших героев в поисковой системе и поспешил вызвать патрульных.

Тем временем Теренс предложил Рiku ознакомиться с книгой известного автора Врийта "Трактат об инструментальном космоанализе". Рiku книга определенно показалась знакомой, особенно его привлекла строка s . Смысла самой строки, он, к сожалению, не понимал, однако в ее частях он видел что-то знакомое. Чтобы разобраться подробнее, Рик решил изучить все подстроки s . Однако изучать равные подстроки не было смысла, а остальные стоило как-либо систематизировать. Например, расставить их по длине и в алфавитном порядке. Поэтому Рик попросил вас узнать, сколько у данной строки существует пар подстрок s_1 и s_2 равной длины, таких, что $s_1 < s_2$ лексикографически.

Формат входных данных

Задана строка s , состоящая из строчных латинских букв ($|s| \leq 2500$).

Формат выходных данных

Выведите одно число — количество искомых пар подстрок.

Пример

стандартный ввод	стандартный вывод
abac	9

Замечание

Рассмотрим подстроки длины 1. Имеется две подстроки "a", каждая из которых меньше подстрок "b" и "c". Также подстрока "b" меньше подстроки "c". Отсюда получаем 5 пар искомых подстрок.

Теперь рассмотрим подстроки длины 2. Подстрока "ab" меньше подстрок "ba" и "ac", а строка "ac" меньше, чем строка "ba". Отсюда получаем еще 3 пары.

Наконец, рассмотрим подстроки длины 3. Подстрока "aba" меньше подстроки "bac".

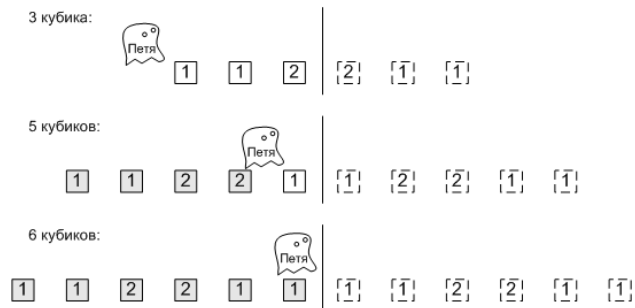
Таким образом, суммарно получаем 9 искомых пар подстрок.

Задача G. Кубики

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

Привидение Петя любит играть со своими кубиками. Он любит выкладывать их в ряд и разглядывать своё творение. Однако недавно друзья решили подшутить над Петей и поставили в его игровой комнате зеркало. Ведь всем известно, что привидения не отражаются в зеркале! А кубики отражаются.

Теперь Петя видит перед собой N цветных кубиков, но не знает, какие из этих кубиков настоящие, а какие — всего лишь отражение в зеркале.



Помогите Пете! Выясните, сколько у него может быть кубиков. Петя видит отражение всех кубиков в зеркале и часть кубиков, которая находится перед ним. Часть кубиков может быть позади Пети, их он не видит.

Формат входных данных

Первая строка входного файла содержит два целых числа: N ($1 \leq N \leq 100\,000$) и количество различных цветов, в которые могут быть раскрашены кубики, — M ($1 \leq M \leq 100\,000$). Следующая строка содержит N целых чисел от 1 до M — цвета кубиков.

Формат выходных данных

В выходной файл выведите в порядке возрастания все такие K , что у Пети может быть K кубиков.

Пример

стандартный ввод	стандартный вывод
6 2 1 1 2 2 1 1	3 5 6

Задача Н. Название команды

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

Во Всеберляндской олимпиаде по программированию могут участвовать команды в составе n человек. Для участия каждой команде нужно выбрать название.

Участники одной из команд составили название, включающее каждое из их имен, и записали его в строку t длины m . Однако, такое название получилось слишком длинным! Поэтому они хотят сократить его следующим образом:

- выбрать префикс строки t минимальной длины, который включал бы все их имена без пересечений.

Префиксом строки называется строка, полученная удалением нескольких (возможно, нуля) последних символов из исходной строки.

Некоторое множество подстрок входит в строку *без пересечений*, если никакой символ не принадлежит двум подстрокам одновременно. Например, подстроки «a» и «bc» входят в строку «ababc» без пересечений, а подстроки «aba» и «abc» — нет.

Если название команды возможно сократить — выведите длину минимального подходящего префикса.

Формат входных данных

Первая строка входных данных содержит два целых числа: n ($2 \leq n \leq 8$) — количество участников в команде, и m ($2 \leq m \leq 2 \cdot 10^5$) — длину названия, составленного командой.

Вторая строка входных данных содержит строку t , состоящую из строчных латинских букв — название, составленное командой.

Далее следуют n строк, каждая из которых содержит строку s_i , состоящую из строчных латинских букв ($1 \leq |s_i| < m$) — имя i -го участника команды ($1 \leq i \leq n$). Длина строки s_i обозначается как $|s_i|$.

Гарантируется, что сумма значений $|s_i|$ не превосходит значения m , и в строку t можно поместить имена всех участников без пересечений.

Формат выходных данных

Выведите:

- Число -1 , если название команды невозможно сократить;
- Иначе длину минимального префикса строки t , в который могут войти имена всех участников без пересечений.

Примеры

стандартный ввод	стандартный вывод
4 22 omihaakrissashaanna kris miha sasha anna	20
2 16 coolnickgoodalex nick alex	-1
3 7 bababab a bab b	5

Задача I. XORофикатор

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

Вам дана бинарная (состоящая из 0 и 1) матрица размера $n \times m$. Также вам дан XORофикатор, который за одно применение умеет инвертировать все значения в выбранной строке (заменяет 0 на 1 и 1 на 0).

Будем называть столбец матрицы *особенным*, если в нём ровно одна 1. Вам надо найти максимальное количество столбцов, которые можно сделать *особенными* одновременно, и сказать, в каких строках нужно для этого применить XORофикатор.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке находится одно целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит два числа n и m ($1 \leq n, m \leq 3 \cdot 10^5$, $n \cdot m \leq 3 \cdot 10^5$).

Каждая из следующих n строк набора содержит бинарную строку длины m .

Гарантируется, что сумма $n \cdot m$ по всем наборам входных данных не превышает $3 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных, выведите две строки.

В первой строке выведите максимальное количество *особенных* столбцов, которых можно получить одновременно.

Во второй строке выведите бинарную строку размера n , где на i -й позиции стоит 0, если XORофикатор на i -й строке применять не нужно, и 1, если XORофикатор на i -й строке применять нужно.

Если есть несколько подходящих конфигураций XORофикатора, вы можете вывести любую из них.

Пример

стандартный ввод	стандартный вывод
5	3
3 4	010
1010	1
0110	0
0100	1
1 1	1
1	3
1 1	00
0	2
2 5	010
00101	
10110	
3 3	
101	
111	
000	

Замечание

В первом наборе входных данных можно применить XORофикатор ко второму ряду, тогда 2-й, 3-й и 4-й столбцы будут *особенными*.

Во втором наборе входных данных единственный столбец уже *особенный*, поэтому применять XORофикатор не надо.

Задача J. Сколько строк меньше

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Дан набор D из n строк и строка s . Требуется быстро находить число строк, лексикографически меньших s , в наборе D .

С заданной строкой s выполняются q модификаций, каждая из которых задается парой из числа k_i и символа c_i . Модификация (k_i, c_i) заключается в том, что все символы строки s , начиная с k_i и до конца строки, заменяются на символ c_i .

Например, пусть исходно строка s была равна «anatoly», тогда последовательность запросов $(5, o)$, $(3, b)$, $(7, x)$ будет менять строку следующим образом:

«anatoly» → «anatooo» → «anbbbb» → «anbbbxb»

После каждого изменения строки s требуется вывести количество строк в наборе D , которые лексикографически меньше, чем s .

Замечание

Строка a лексикографически меньше строки b , если $a \neq b$ и выполнено одно из двух условий:

- a является префиксом строки b ;
- для некоторого i первые i символов строки a равны соответствующим символам строки b , а $a_{i+1} < b_{i+1}$.

Формат входных данных

Первая строка входных данных содержит два целых числа n и q — количество строк в наборе D и количество модификаций ($1 \leq n, q \leq 10^6$).

Во второй строке находится строка s , состоящая из не более чем 10^6 строчных латинских букв.

В следующих n строках содержатся строки набора D . Каждая строка состоит из строчных латинских букв. Суммарная длина строк в D не превосходит 10^6 .

Следующие q строк содержат описания модификаций. Описание состоит из числа k_i и строчной буквы латинского алфавита c_i , разделенных пробелом ($1 \leq k_i \leq |s|$).

Формат выходных данных

В первой строке выведите число строк в наборе D , которые лексикографически меньше исходной строки s .

Затем выведите q строк. В i -й строке выведите ответ после i -й модификации.

Примеры

стандартный ввод	стандартный вывод
4 3	0
anatoly	0
boris	2
anatooo	3
anbbbbu	
anba	
5 o	
3 b	
7 x	
5 5	3
abcde	3
buz	3
ababa	4
build	4
a	1
aba	
1 b	
3 z	
2 u	
4 z	
1 a	

Замечание

В первом тесте из примера строка изменяется следующим образом:

«anatoly» → «anatooo» → «anbbbbb» → «anbbbbx».

- Изначальная строка «anatoly» лексикографически меньше всех строк набора, поэтому ответ на задачу 0.
- После первого изменения строка становится «anatooo» и такая строка есть в наборе, однако ответ на задачу по прежнему будет 0, так как она не меньше, текущей.
- Затем строка становится «anbbbbb», что лексикографически больше, чем «anatooo» и «anba», но меньше чем «anbbbbu» и «boris», таким образом ответ 2.
- После последнего изменения строка станет «anbbbbx», что лексикографически больше «anatooo», «anba» и «anbbbbu», ответ 3.

Задача К. Яся и таинственное дерево

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	512 мегабайт

Яся гуляла по лесу и совершенно случайно нашла дерево на n вершинах. Дерево — это связный неориентированный граф, в котором отсутствуют циклы.

Рядом с деревом девочка нашла древний манускрипт, на котором записаны m запросов. Запросы бывают двух видов.

Первый вид запросов описывается числом y . Вес **каждого** ребра в дереве заменяется на побитовое исключающее «ИЛИ» веса этого ребра и числа y .

Второй вид описывается вершиной v и числом x . Яся выбирает вершину u ($1 \leq u \leq n$, $u \neq v$) и мысленно проводит в дереве двунаправленное ребро веса x из v в u .

Затем Яся находит простой цикл в получившемся графе и считает побитовое исключающее «ИЛИ» от всех рёбер на нём. Она хочет выбрать такую вершину u , чтобы посчитанное значение было **максимально**. Это посчитанное значение и будет ответом на запрос. Можно показать существование и единственность такого цикла в указанных ограничениях (независимо от выбора u). Если ребро между v и u уже существовало, простым циклом будет путь $v \rightarrow u \rightarrow v$.

Обратите внимание, что запрос второго типа выполняется *мысленно*, то есть дерево после него **никак** не меняется.

Помогите Ясе ответить на все запросы.

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Далее следуют описания наборов.

В первой строке каждого набора даны целые числа n , m ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 2 \cdot 10^5$) — количество вершин в дереве и количество запросов.

В следующих $n-1$ строках каждого набора даны целые числа v , u , w ($1 \leq v, u \leq n$, $1 \leq w \leq 10^9$) — концы некоторого ребра в дереве и его вес.

Гарантируется, что заданный набор рёбер образует дерево.

В следующих m строках каждого набора описаны запросы:

- $\wedge y$ ($1 \leq y \leq 10^9$) — параметр запроса первого типа;
- $? v x$ ($1 \leq v \leq n$, $1 \leq x \leq 10^9$) — параметры запроса второго типа.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$. То же самое гарантируется для m .

Формат выходных данных

Для каждого набора входных данных выведите ответы на запросы второго типа.

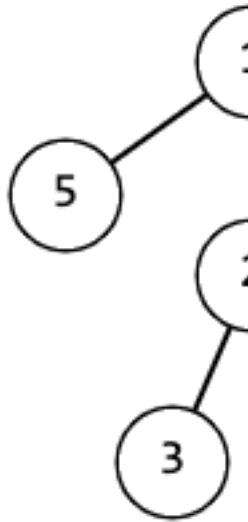
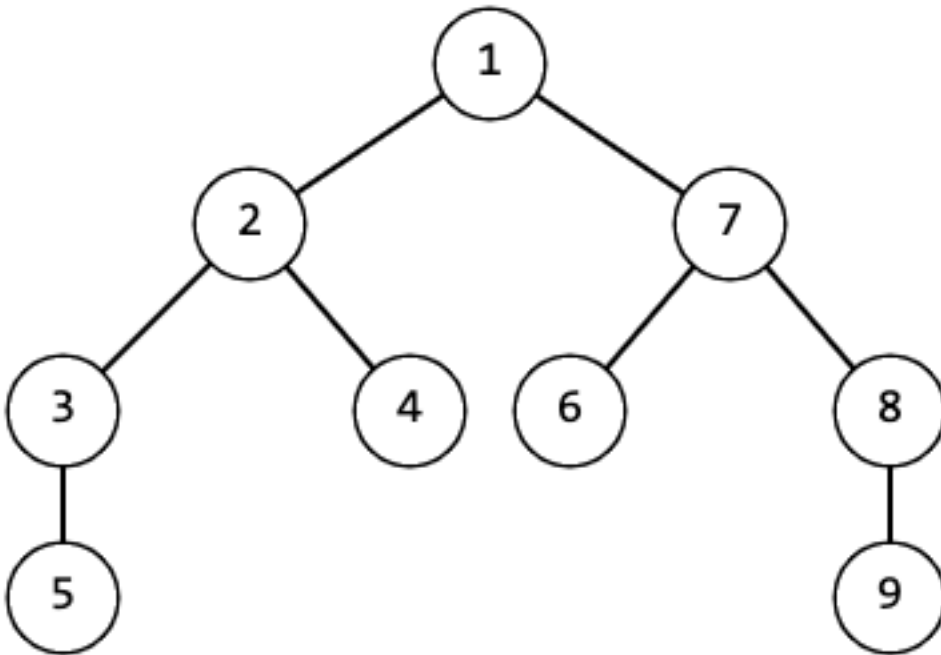
Примеры

стандартный ввод	стандартный вывод
2 3 7 1 2 1 3 1 8 ~ 5 ? 2 9 ~ 1 ? 1 10 ~ 6 ? 3 1 ? 2 9 5 6 1 2 777 3 2 2812 4 1 16 5 3 1000000000 ~ 4 ? 3 123 ? 5 1000000000 ~ 1000000000 ? 1 908070 ? 2 1	13 15 11 10 1000000127 2812 999756331 999999756
3 8 4 8 6 3 6 3 4 2 5 4 7 6 2 7 1 10 4 1 4 5 1 2 ~ 4 ~ 7 ? 7 8 ? 4 10 5 6 3 1 4 2 3 9 4 3 6 5 2 10 ? 5 7 ~ 1 ~ 8 ? 4 10 ? 1 9 ? 3 6 4 2 2 1 4 4 3 5 2 3 4 ~ 13 ? 1 10	14 13 13 8 11 11 10

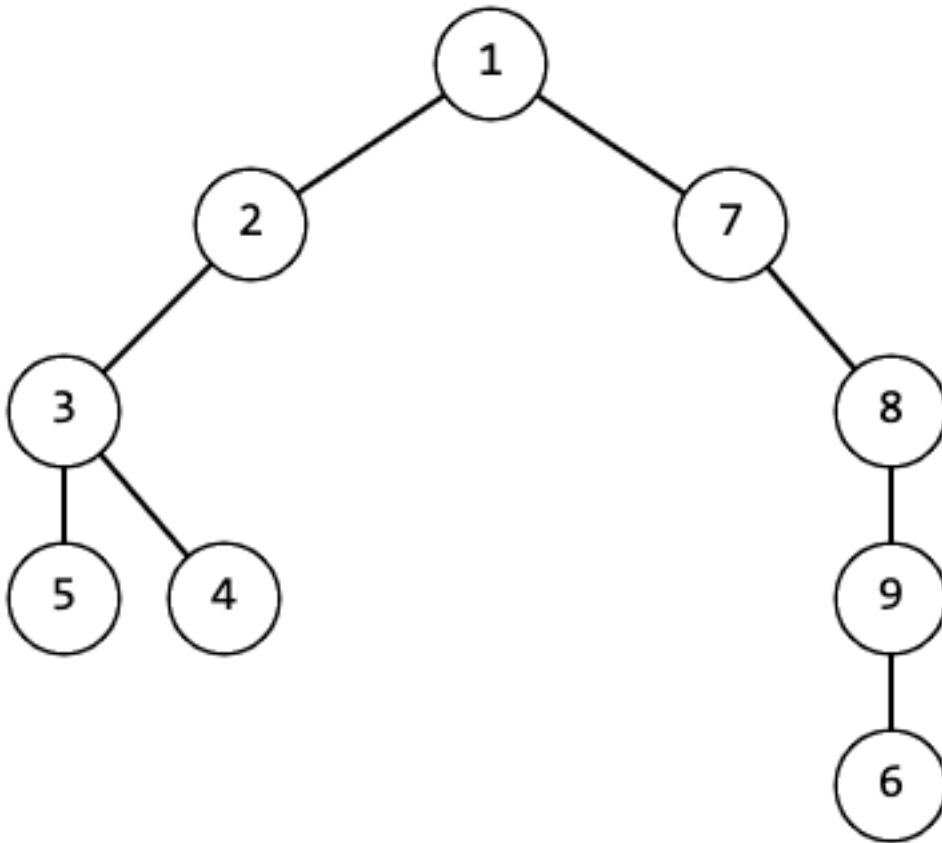
Задача L. Симметрия

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

Киду подарили дерево из n вершин с корнем в вершине 1. Так как он очень любит *симметричные* объекты, Кид хочет узнать *симметрично* ли это дерево.



Например, деревья на картинке выше *симметричны*.



А деревья на этой картинке не *симметричны*.

Более формально, дерево является *симметричным*, если существует такой порядок детей, что:

- Поддерево самого левого ребёнка корня является зеркальным отражением поддерева самого правого ребёнка;
- поддерево второго слева ребёнка корня является зеркальным отражением поддерева второго справа ребёнка корня;
- ...
- если число детей корня нечётно, то поддерево среднего ребёнка должно быть *симметрично*.

Формат входных данных

Первая строка входных данных содержит единственное число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Первая строка каждого набора содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве.

Следующие $n - 1$ строк содержат по два числа u и v ($1 \leq u, v \leq n, u \neq v$) — номера вершин, соединённых ребром.

Гарантируется, что сумма n по всем наборам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Выведите t строк, каждая из которых является ответом на соответствующий набор входных данных. В качестве ответа выведите «YES», если данное дерево *симметрично*, и «NO» в противном случае.

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Пример

стандартный ввод	стандартный вывод
6	YES
6	NO
1 5	YES
1 6	NO
1 2	NO
2 3	YES
2 4	
7	
1 5	
1 3	
3 6	
1 4	
4 7	
4 2	
9	
1 2	
2 4	
2 3	
3 5	
1 7	
7 6	
7 8	
8 9	
10	
2 9	
9 10	
2 3	
6 7	
4 3	
1 2	
3 8	
2 5	
6 5	
10	
3 2	
8 10	
9 7	
4 2	
8 2	
2 1	
4 5	
6 5	
5 7	
1	