

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

## Задача С. 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 D. Гений Евгенийевич

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

Гений Евгенийевич давно мечтал о дереве. Сейчас в моде деревья с  $n$  вершинами, на каждой из которых написано число  $x_i$ . Долгожданный день настал, и Гений Евгенийевич смог себе приобрести такое дерево. Вернувшись из магазина после покупки, он начал считать различные величины на этом дереве. Все ему давалось очень просто, пока он не решил найти путь в этом дереве, значение величины  $l(p(u, v)) \cdot \min(x_u, x_v)$  для которого максимально. Здесь  $p(u, v)$  — путь между вершинами  $u$  и  $v$ ,  $l(p)$  — длина пути  $p$  в ребрах.

Уже который день Гений Евгенийевич не выходит на улицу. Его друзья, естественно, начали волноваться. Они просят Вас помочь Гению Евгенийевичу справиться с задачей.

### Формат входных данных

В первой строке входных данных дано число  $n$  ( $1 \leq n \leq 10^5$ ) — количество вершин в дереве.

Во второй строке даны  $n$  чисел  $x_i$  ( $1 \leq x_i \leq 10^9$ ) — числа написанные на вершинах. На  $i$ -й вершине написано число  $x_i$ .

В следующих  $n - 1$ -й строке идут пары чисел  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — рёбра в дереве.

### Формат выходных данных

Выведите максимальное значение величины, описанной в условии.

### Пример

стандартный ввод	стандартный вывод
8	21
9 2 8 8 7 2 1 2	
2 1	
3 1	
4 1	
5 2	
6 4	
7 5	
8 5	

## Задача Е. 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 F. Интерактивная вершина

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

### Это интерактивная задача

У Ильдара есть дерево из  $n$  вершин и он показал его вам. Он выбирает одну вершину  $u$  как специальную вершину, но он не сообщает вам ничего о ней!

Вместо этого, вы можете задавать ему вопросы. Для каждого вопроса вы должны выбрать вершину  $x$ , натуральное число  $k$  и  $k$  вершин  $v_1, v_2, \dots, v_k$  и он вам скажет, правда ли, что  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ . Здесь,  $\text{dist}(p, q)$  это количество ребер на простом пути между вершинами  $p$  и  $q$  в дереве.

Вы должны угадать специальную вершину за не больше, чем  $4 \lceil \log_2 n \rceil$  вопросов.

Ильдар очень добрый, поэтому он не будет менять специальную вершину между вашими вопросами (другими словами, интерактор неадаптивный).

**Поскольку ограничения большие и flush это тяжелая операция, убедитесь, что вы не делаете операцию flush слишком часто. Рекомендуется делать flush только после вывода каждого вопроса.**

### Протокол взаимодействия

Процесс взаимодействия начинается с того, что на первой строке вводится целое число  $n$ : количество вершин в дереве Ильдара ( $2 \leq n \leq 200\,000$ ).

Каждая из следующих  $n - 1$  строк содержит два целых числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ), означающих ребро между  $u$  и  $v$ . Гарантируется, что данные ребра образуют дерево.

После этого, вы можете задавать вопросы.

Чтобы задать вопрос, выведите одну строку, содержащую “?  $k$ ” ( $1 \leq k \leq n$ ), целое число  $x$  ( $1 \leq x \leq n$ ) и затем  $k$  различных целых чисел  $v_1, v_2, \dots, v_k$  ( $1 \leq v_i \leq n$ ). Разделяйте соседние числа в строке ровно одним пробелом. Затем сделайте flush выходного потока.

После каждого вопроса, считайте одно целое число  $ans \in \{0, 1\}$ . Если  $\min(\text{dist}(u, v_i)) \geq \text{dist}(u, x)$ , тогда  $ans$  будет равно 1. Иначе,  $ans$  будет равно 0.

Когда вы нашли специальную вершину  $u$  ( $1 \leq u \leq n$ ), выведите одно целое число “!  $u$ ”, сделайте flush выходного потока и завершите работу программы.

Ваше решение получит Wrong Answer или Time Limit Exceeded если вы сделаете больше чем  $4 \lceil \log_2 n \rceil$  вопросов.

Ваше решение получит Idleness Limit Exceeded если оно не совершает никаких действий или не делает flush выходного потока.

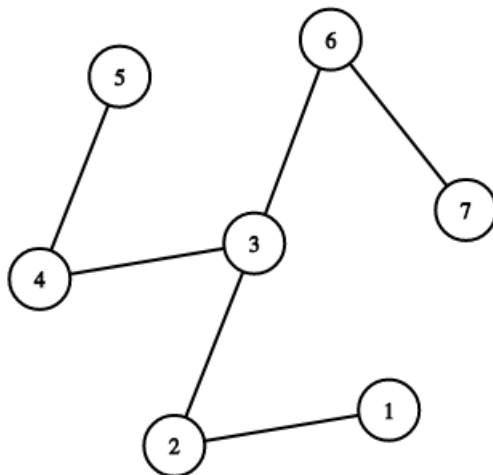
Чтобы сделать flush выходного потока, вы можете использовать:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- используйте документацию других языков.

## Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 1 2 1 3 1 4 1 5 1	? 4 1 2 3 4 5 ! 1
5 1 2 1 3 1 4 1 5 0 0 0 0	? 4 1 2 3 4 5 ? 3 1 2 3 4 ? 2 1 2 3 ? 1 1 2 ! 2
7 1 2 2 3 3 4 4 5 3 6 6 7 1	? 3 3 5 7 1 ! 3

## Замечание





## Задача GBonus. Циклы в произведении

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	256 мегабайт

Рассмотрим дерево (то есть неориентированный связный граф без циклов)  $T_1$  и дерево  $T_2$ . Определим их *декартово произведение*  $T_1 \times T_2$  следующим образом.

Пусть  $V$  — множество вершин  $T_1$ , а  $U$  — множество вершин  $T_2$ .

Тогда множеством вершин  $T_1 \times T_2$  является  $V \times U$ , то есть множество упорядоченных пар вершин, где первая вершина из  $V$ , а вторая из  $U$ .

Проведём ребра следующего вида:

- Между  $(v, u_1)$  и  $(v, u_2)$  есть неориентированное ребро, если  $u_1$  и  $u_2$  смежны в графе  $U$ .
- Аналогично между  $(v_1, u)$  и  $(v_2, u)$  есть неориентированное ребро, если  $v_1$  и  $v_2$  смежны в графе  $V$ .

Обратите внимание на пояснения к примерам, они содержат картинки для произведения деревьев для тестов из примеров.

Рассмотрим граф  $T_1 \times T_2$ . Вычислите количество циклов в этом графе (не обязательно простых) длины  $k$ . Выведите остаток при делении найденного количества на 998244353.

Циклом называется последовательность вершин  $w_1, w_2, \dots, w_k$ , где  $w_i \in V \times U$ , такая что любые две соседние вершины смежны, а также  $w_1$  смежно с  $w_k$ . Циклы отличающиеся только циклическим сдвигом или направлением обхода всё равно считаются **различными**.

### Формат входных данных

В первой строке входного файла даны три числа —  $n_1, n_2$  и  $k$  ( $2 \leq n_1, n_2 \leq 4000, 2 \leq k \leq 75$ ) — размер первого дерева, второго дерева и длина цикла соответственно.

Затем следует  $n_1 - 1$  строка описывающая первое дерево. Каждая из этих строк содержит два числа —  $v_i, u_i$  ( $1 \leq v_i, u_i \leq n_1$ ), задающие соответствующее ребро графа.

Каждая из следующих  $n_2 - 1$  строк задаёт второе дерево в том же формате.

Гарантируется, что заданные графы являются деревьями.

### Формат выходных данных

Выведите одно число — количество циклов по модулю 998244353.

### Примеры

стандартный ввод	стандартный вывод
2 2 2 1 2 1 2	8
2 2 4 1 2 1 2	32
2 3 4 1 2 1 2 1 3	70
4 2 2 1 2 1 3 1 4 1 2	20

## Замечание

Следующие три картинки иллюстрируют графы, которые получаются в результате произведения в тестах из условия.

В первом примере список циклов длины 2 следующий:

- «AB», «BA»
- «BC», «CB»
- «AD», «DA»
- «CD», «DC»

