

## Задача А. Выбор вершин дерева

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

Дан граф, являющийся деревом. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества посчитаем количество вершин в нём. Каково максимальное из этих количеств?

### Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины  $i$ , не являющейся корнем, задан номер вершины-предка  $p_i$  в корневом дереве. Дерево, заданное таким образом, состоит из рёбер  $i-p_i$  для всех вершин  $i$ , кроме корня.

В первой строке записано целое число  $n$  — количество вершин в графе ( $1 \leq n \leq 100$ ). В следующих  $n$  строках задан граф. В  $i$ -й из этих строк записано целое число  $p_i$  — номер вершины-предка  $i$ -й вершины. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ .

Гарантируется, что заданный граф является деревом.

### Формат выходных данных

В первой строке выведите одно число — максимальное количество вершин в допустимом множестве.

## Примеры

стандартный ввод	стандартный вывод	пояснение
5 0 1 1 2 3	3	
6 5 6 5 1 0 1	3	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальным количеством вершин в нём.

## Задача В. Выбор вершин взвешенного дерева

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

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

### Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины  $i$ , не являющейся корнем, задан номер вершины-предка  $p_i$  в корневом дереве. Дерево, заданное таким образом, состоит из рёбер  $i - p_i$  для всех вершин  $i$ , кроме корня.

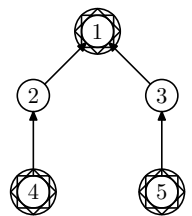
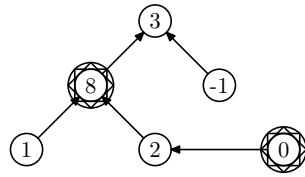
В первой строке записано целое число  $n$  — количество вершин в графе ( $1 \leq n \leq 100$ ). В следующих  $n$  строках задан граф. В  $i$ -й из этих строк записаны через пробел два целых числа  $p_i$  и  $q_i$ ; здесь  $p_i$  — номер вершины-предка  $i$ -ой вершины, а  $q_i$  — число, записанное в этой вершине. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ . Числа  $q_i$  не превосходят 10 000 по абсолютной величине.

Гарантируется, что заданный граф является деревом.

### Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве.

## Примеры

стандартный ввод	стандартный вывод	пояснение
5 0 1 1 2 1 3 2 4 3 5	10	
6 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальной суммой чисел в вершинах.

### Задача С. Ограниченный выбор вершин взвешенного дерева

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

Дан ориентированный граф, являющийся корневым деревом. Дуги графа направлены от сыновей к отцам, и из каждой вершины можно по дугам прийти в корень. В вершинах графа написаны целые числа.

Множество вершин графа называется *недопустимым*, если оно содержит более  $k$  сыновей какой-либо вершины.

Множество вершин графа называется *допустимым*, если оно не является недопустимым, а кроме того, никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

#### Формат входных данных

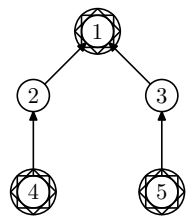
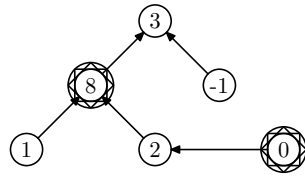
В первой строке записаны через пробел два целых числа  $n$  и  $k$  — количество вершин в графе и ограничение на количество взятых сыновей какой-либо вершины, соответственно ( $1 \leq n \leq 100$ ,  $0 \leq k \leq n$ ). В следующих  $n$  строках задан граф. В  $i$ -й из этих строк записаны через пробел два целых числа  $p_i$  и  $q_i$ ; здесь  $p_i$  — номер вершины-предка  $i$ -ой вершины, а  $q_i$  — число, записанное в этой вершине. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ . Числа  $q_i$  не превосходят 10 000 по абсолютной величине.

Гарантируется, что заданный граф является корневым деревом.

#### Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве.

### Примеры

стандартный ввод	стандартный вывод	пояснение
5 2 0 1 1 2 1 3 2 4 3 5	10	
6 1 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество с максимальной суммой чисел в вершинах.

### Задача D. Выбор $k$ вершин взвешенного дерева

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

Дан граф, являющийся деревом. В вершинах графа написаны целые числа. Множество вершин графа называется *допустимым*, если никакие две вершины этого множества не соединены ребром.

Рассмотрим все допустимые множества вершин графа, состоящие ровно из  $k$  вершин. Для каждого такого множества вычислим сумму чисел, написанных в его вершинах. Какова максимальная из этих сумм?

#### Формат входных данных

Граф в этой задаче задан в виде *корневого дерева*. В графе выделена вершина — *корень дерева*. Для каждой вершины  $i$ , не являющейся корнем, задан номер вершины-предка  $p_i$  в корневом дереве. Дерево, заданное таким образом, состоит из рёбер  $i - p_i$  для всех вершин  $i$ , кроме корня.

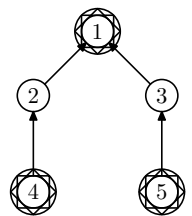
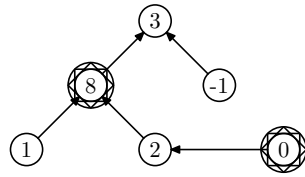
В первой строке записаны через пробел два целых числа  $n$  и  $k$  — количество вершин в графе и размер множеств, соответственно ( $1 \leq n \leq 100$ ,  $0 \leq k \leq n$ ). В следующих  $n$  строках задан граф. В  $i$ -й из этих строк записаны через пробел два целых числа  $p_i$  и  $q_i$ ; здесь  $p_i$  — номер вершины-предка  $i$ -ой вершины, а  $q_i$  — число, записанное в этой вершине. Для корня дерева  $p_i = 0$ ; для всех остальных вершин  $1 \leq p_i \leq n$ . Числа  $q_i$  не превосходят 10 000 по абсолютной величине.

Гарантируется, что заданный граф является деревом, а также что в нём существует хотя бы одно допустимое множество, состоящее ровно из  $k$  вершин.

#### Формат выходных данных

В первой строке выведите одно число — максимальную сумму чисел в допустимом множестве из  $k$  вершин.

### Примеры

стандартный ввод	стандартный вывод	пояснение
5 3 0 1 1 2 1 3 2 4 3 5	10	
6 2 5 8 6 0 5 -1 1 1 0 3 1 2	8	

На рисунках показаны графы, заданные в примерах. В каждом графе выделено допустимое множество из  $k$  вершин с максимальной суммой чисел в них.

## Задача Е. Раскраска дерева

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

Дано корневое дерево из  $n$  вершин. Корень дерева имеет ровно три сына. Все вершины, за исключением корня и листьев, имеют ровно по два сына.

Все листья дерева покрашены в один из трёх цветов. Необходимо раскрасить остальные вершины дерева в эти же три цвета так, чтобы смежные вершины были окрашены в различные цвета, либо определить, что такой раскраски не существует.

### Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. Каждый тестовый случай задаётся следующим образом.

Сначала в отдельной строке записано целое число  $n$  ( $4 \leq n \leq 300\,000$ ). В следующих  $n - 1$  строках описывается дерево. В  $i$ -й из этих строк задано одно целое число  $p$  ( $1 \leq p \leq i$ ) — номер отца вершины с номером  $i + 1$ . Корень дерева всегда имеет номер 1, а остальные вершины пронумерованы от 2 до  $n$ .

Далее в отдельной строке дано целое число  $m$  — количество листьев данного дерева. В следующих  $m$  строках перечислены листья дерева, по одному в строке. Описание каждого листа состоит из его номера (целое число от 2 до  $n$ ) и цвета (целое число от 1 до 3).

Гарантируется, что сумма всех  $n$  во входных данных также не превосходит 300 000. После описания каждого тестового случая следует пустая строка. Входные данные завершаются строкой, на которой вместо  $n$  записано число 0.

### Формат выходных данных

В ответ на каждый тестовый случай выведите в отдельной строке «YES», если раскраска существует, и «NO» в противном случае. Если раскраска существует, то в следующей строке выведите  $n$  чисел. В этой строке  $i$ -е число должно задавать цвет вершины с номером  $i$ . Если раскрасок существует несколько, то можно вывести любую из них.

## Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 1 1 1 3 2 3 3 1 4 2  0	NO
6 1 1 1 2 2 4 5 2 6 3 3 1 4 1  0	YES 2 1 1 1 2 3

## Задача F. Красно-чёрное дерево

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

Рассмотрим двоичное корневое дерево: в этом дереве выделена вершина-корень, все рёбра ведут в направлении от корня, и у каждой вершины от нуля до двух детей. Будем считать, что из каждой вершины выходят ровно два ребра: если при этом у неё меньше двух детей, оставшиеся рёбра ведут в пустоту.

Будем называть дерево *красно-чёрным*, если выполнены следующие условия:

- Каждая вершина покрашена либо в красный, либо в чёрный цвет.
- В дереве нет ребра с двумя красными концами. Пустота считается чёрной.
- Рассмотрим все пути по рёбрам, идущие из корня в пустоту. Количество чёрных вершин на всех таких путях одинаково.

Аналогичную раскраску можно использовать в двоичных деревьях поиска для их балансировки.

Дано непустое двоичное корневое дерево. Покрасьте его так, чтобы оно стало красно-чёрным, или выясните, что это невозможно.

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 500$ ) — количество вершин в дереве. Вершины пронумерованы числами от 1 до  $n$ .

В следующей строке записаны через пробел  $n$  целых чисел:  $p_1, p_2, \dots, p_n$  ( $0 \leq p_i \leq n$ ). Число  $p_i > 0$  означает, что вершина  $i$  — ребёнок вершины  $p_i$ . Если же  $p_i = 0$ , то  $i$  — корень дерева.

Гарантируется, что во входных данных корректно задано двоичное корневое дерево: корень ровно один, у каждой вершины от нуля до двух детей, а кроме того, из корня можно, двигаясь по рёбрам, попасть во все остальные вершины.

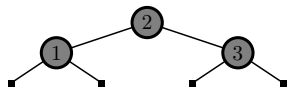
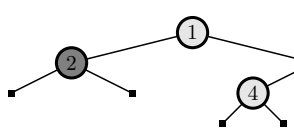
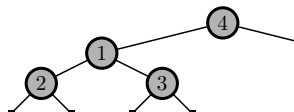
### Формат выходных данных

Если можно покрасить заданное дерево так, чтобы оно стало красно-чёрным, выведите любую такую раскраску в виде строки из  $n$  символов.

Символ на  $i$ -й позиции должен быть равен «R», если вершина  $i$  красная, и «B», если она чёрная.

Если же покрасить дерево невозможно, выведите слово «Impossible».

### Примеры

стандартный ввод	стандартный вывод	пояснение
3 2 0 2	BBB	
4 0 1 1 3	RBRR	
4 4 1 1 0	Impossible	

## Задача G. Редукция дерева

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

Задано неориентированное дерево, содержащее  $n$  вершин. Можно выбрать некоторое ребро и удалить его, при этом инцидентные ему вершины не удаляются. Таким образом можно удалить из дерева некоторый набор рёбер. В результате дерево распадается на некоторое количество меньших деревьев. Требуется, удалив наименьшее количество рёбер, получить в качестве хотя бы одной из компонент связности дерево, содержащее ровно  $p$  вершин.

### Формат входных данных

Первая строка содержит пару натуральных чисел  $n$  и  $p$  ( $1 \leq p \leq n \leq 1000$ ). Далее в  $n - 1$  строке содержатся описания рёбер дерева. Каждое описание состоит из пары натуральных чисел  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n$ ) — номеров соединяемых ребром вершин.

### Формат выходных данных

В первую строку выведите наименьшее количество рёбер  $q$  в искомом наборе. Во вторую строку выведите номера удаляемых рёбер. Номера рёбер определяются порядком их задания по входном файле. Рёбра нумеруются с единицы. Если оптимальных решений несколько, разрешается выводить любое.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
11 6	2
1 2	3 6
1 3	
1 4	
2 6	
2 7	
1 5	
2 8	
4 9	
4 10	
4 11	

**Задача Н. Простые пути в дереве**

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

Дан неориентированный связный граф из  $n$  вершин и  $n - 1$  ребра. Требуется для каждого ребра посчитать суммарную длину простых путей, проходящих через это ребро. Длиной пути здесь называется количество рёбер в пути.

**Формат входных данных**

В первой строке задано целое число  $n$  ( $2 \leq n \leq 300\,000$ ). Следующие  $n - 1$  строк содержат пары чисел от 1 до  $n$  — рёбра графа.

**Формат выходных данных**

Выведите  $n - 1$  строку:  $i$ -я строка должна содержать целое число — ответ для  $i$ -го ребра.

**Пример**

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5	13
1 2	8
2 3	8
2 4	9
5 1	

## Задача I. Демократия

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

В Котийской Республике процветает демократия: как главный демократ сказал, так и будет. В Республике есть  $n$  городов и  $k$  политических партий. Руководителем в каждом городе является представитель одной из партий.

Также в Республике широкое развитие получили информационные и нанотехнологии, в особенности их альянс — новейшие локальные сети с нанотрафиком. Уже построены некоторые каналы связи между городами с использованием этой технологии, причём все города связаны (возможно, через промежуточные города). Технология слишком нова, поэтому набор каналов, необходимых для этого, минимален по количеству.

Главный демократ постановил провести всеобщий политический форум. С этой целью, кроме введения комендантского часа, было решено также выбрать  $k$  городов — по одному с руководителем из каждой партии, — а также построить дополнительные каналы связи, чтобы информация могла быть доставлена между выбранными городами, не проходя через города, не выбранные для форума. А для обеспечения полнейшей безопасности все каналы, ведущие к городам, не участвующим в форуме, должны быть разобраны и разворованы.

Официальное открытие форума, как всегда, состоялось вчера, поэтому требуется проложить каналы как можно быстрее. Государственный монополист, компания «КОТ и Ко», способна прокладывать или разворовывать только один канал в один момент времени, поэтому время, которое требуется для завершения процесса, равно сумме времён прокладок и разворываний каналов. Новейшие разработки в области информационных нанотехнологий позволяют построить один новый канал всего за одну минуту. Время, необходимое для разборки и разворывания канала, также равно одной минуте, хотя нанотехнологии в этом случае ни при чём.

Посчитайте минимальное время, через которое поставленная задача может быть выполнена, а также выясните, какие города следует выбрать для форума и какие каналы связи нужно построить и разобрать.

### Формат входных данных

В первой строке заданы целые числа  $n$  и  $k$  — количество городов и политических партий в Республике ( $1 \leq n \leq 2000$ ,  $1 \leq k \leq 10$ ).

Во второй строке записано  $n$  целых чисел от 1 до  $k$  — для каждого города номер партии, которой он принадлежит. Каждой партии принадлежит хотя бы один город.

Далее в  $(n - 1)$  строке описаны существующие каналы связи: каждая из этих строк содержит два числа  $a_i$  и  $b_i$ , означающих, что существует канал связи между городами  $a_i$  и  $b_i$ . Города пронумерованы целыми числами от 1 до  $n$ .

### Формат выходных данных

В первой строке выведите единственное целое число — минимально возможное время в минутах для завершения процесса. Далее выведите по одной строке для каждой минуты. Строка должна начинаться словом «build», если нужно построить канал, или словом «destroy», если канал нужно разорвать. Далее должны следовать два числа — номера городов, между которыми прокладывается или разбирается канал. В последней строке выведите  $k$  чисел — номера городов, участвующих в форуме. Если решений, на которые требуется минимальное время, несколько, можно вывести любое из них.

### Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
7 3	3
2 1 1 1 1 1 3	build 6 1
1 2	destroy 1 2
2 3	destroy 5 6
3 4	1 6 7
4 5	
5 6	
6 7	

**Задача J. Подготовка к игре**

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

«Трое играющих берут четыре фишки, причём пятый игрок всё время выкидывает. После того, как лиса оказывается съеденной, она делает четыре хода назад.»

кинофильм «Подкидыш»

А теперь пора подготовить игровое поле. На нём  $n$  позиций, некоторые из которых соединены с другими. Соединения устроены так, что существует ровно один путь между каждой парой позиций. Также есть несколько фишек, которые можно расставлять на позициях. Есть лишь одно ограничение: никакие две фишки не могут быть расположены на расстоянии меньше, чем  $c$  ходов.

Чтобы выбрать действительно случайное расположение, найдите для начала число способов расставить фишки. Разрешается расставлять любое число фишек. Или не расставлять их вовсе.

В чём же цель игры, спросите вы? Я не знаю. Может быть, вам и предстоит изобрести её.

**Формат входных данных**

Входные данные состоят из одного или более тестов.

Каждый тест начинается строкой, в которой записаны целое число  $n$  ( $1 \leq n \leq 10\,000$ ), которое определяет число позиций на поле, и целое число  $c$  ( $1 \leq c \leq \min(n, 500)$ ), которое определяет минимальное расстояние между фишками. Следующая  $n - 1$  строка описывает пары полей, которые соединены между собой.

Входные данные завершаются тестом, где  $n = c = 0$ , который не нужно обрабатывать.

Сумма  $n$  по всем тестам во входных данных не превысит 10 000.

**Формат выходных данных**

Для каждого теста выведите искомое число способов. Поскольку это число может быть очень большим, его следует вывести по модулю  $10^6$ .

Следуйте формату вывода, указанному в примере, как можно точнее.

**Пример**

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 2	Case #1: 14
1 2	Case #2: 3
1 3	Case #3: 4
1 4	
4 5	
2 2	
1 2	
2 1	
1 2	
0 0	