

**Задача А. Разрезание графа**

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

Дан неориентированный граф. Над ним в заданном порядке производят операции следующих двух типов:

- `cut` — разрезать граф, то есть удалить из него ребро;
- `ask` — проверить, лежат ли две вершины графа в одной компоненте связности.

Известно, что после выполнения всех операций типа `cut` рёбер в графе не осталось. Найдите результат выполнения каждой из операций типа `ask`.

**Формат входного файла**

Первая строка входного файла содержит три целых числа, разделённые пробелами — количество вершин графа  $n$ , количество рёбер  $m$  и количество операций  $k$  ( $1 \leq n \leq 50\,000$ ,  $0 \leq m \leq 100\,000$ ,  $m \leq k \leq 150\,000$ ).

Следующие  $m$  строк задают рёбра графа;  $i$ -я из этих строк содержит два числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ), разделённые пробелами, — номера концов  $i$ -го ребра. Вершины нумеруются с единицы; граф не содержит петель и кратных рёбер.

Далее следуют  $k$  строк, описывающих операции. Операция типа `cut` задаётся строкой «`cut u v`» ( $1 \leq u, v \leq n$ ), которая означает, что из графа удаляют ребро между вершинами  $u$  и  $v$ . Операция типа `ask` задаётся строкой «`ask u v`» ( $1 \leq u, v \leq n$ ), которая означает, что необходимо узнать, лежат ли в данный момент вершины  $u$  и  $v$  в одной компоненте связности. Гарантируется, что каждое ребро графа встретится в операциях типа `cut` ровно один раз.

**Формат выходного файла**

Для каждой операции `ask` во входном файле выведите на отдельной строке слово «YES», если две указанные вершины лежат в одной компоненте связности, и «NO» в противном случае. Порядок ответов должен соответствовать порядку операций `ask` во входном файле.

**Пример**

cutting.in	cutting.out
3 3 7	YES
1 2	YES
2 3	NO
3 1	NO
ask 3 3	
cut 1 2	
ask 1 2	
cut 1 3	
ask 2 1	
cut 2 3	
ask 3 1	

**Задача В. Расстояние между вершинами**

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

Коль Дейкстру́ писать без кучи,  
 То тайм-лимит ты получишь...  
 А в совсем крутой задаче  
 Юзай кучу Фибоначчи!

Спектакль преподавателей ЛКШ.июль-2007

Дан неориентированный взвешенный граф. Требуется найти вес минимального пути между двумя вершинами.

**Формат входного файла**

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количества вершин и рёбер графа соответственно ( $1 \leq n \leq 100\,000$ ,  $1 \leq m \leq 200\,000$ ). Вторая строка входного файла содержит натуральные числа  $s$  и  $t$  — номера вершин, длину пути между которыми требуется найти ( $1 \leq s, t \leq n$ ,  $s \neq t$ ).

Следующие  $m$  строк содержат описание рёбер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номерами концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $0 \leq w_i \leq 100$ ).

**Формат выходного файла**

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами  $s$  и  $t$ , или -1, если такого пути нет.

**Пример**

distance.in	distance.out
4 4	3
1 3	
1 2 1	
2 3 2	
3 4 5	
4 1 4	

**Задача С. Островные государства — 2**

Имя входного файла: `island2.in`  
 Имя выходного файла: `island2.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

Суровые феодальные времена переживала некогда великая островная страна Байтландия. За главенство над всем островом борются два самых сильных барона. Таким образом,

каждый город страны контролируется одним из правителей. Как водится издревле, некоторые из городов соединены двусторонними дорогами. Бароны очень не любят друг друга и стараются делать как можно больше пакостей. В частности, теперь для того, чтобы пройти по дороге, соединяющей города различных правителей, надо заплатить пошлину — один байтландский рубль. Кроме этого, за выезд из городов с чётными номерами берется удвоенная пошлина.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское собрание программистов. Разумеется, он хочет затратить при этом как можно меньше денег, и помочь ему здесь, как обычно, предлагается вам.

### Формат входного файла

В первой строке входного файла записаны два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество городов и количество дорог соответственно.

В следующей строке содержится информация о городах —  $N$  чисел 1 или 2 — какому из баронов принадлежит соответствующий город.

В последних  $M$  строках записаны пары  $1 \leq a, b \leq N$ ,  $a \neq b$ . Каждая пара означает наличие дороги из города  $a$  в город  $b$ . По дорогам Байтландии можно двигаться в любом направлении.

### Формат выходного файла

Если искомого пути не существует, выведите единственное слово «impossible». В противном случае в первой строке напишите минимальную стоимость и количество посещённых городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Примеры

island2.in	island2.out
7 8 1 1 1 1 2 2 1 1 2 2 5 2 3 5 4 4 3 4 7 1 6 6 7	0 5 1 2 3 4 7
5 5 1 2 1 1 2 1 2 2 3 3 5 1 4 4 5	2 3 1 4 5

### Задача D. День Объединения

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

В Байтландии есть целых  $n$  городов, но нет ни одной дороги. Король страны, Вальдемар де Беар, решил исправить эту ситуацию и соединить некоторые города дорогами так, чтобы по этим дорогам можно было добраться от любого города до любого другого. Когда строительство будет завершено, король планирует отпраздновать День Объединения. К сожалению, казна Байтландии почти пуста, поэтому король требует сэкономить деньги, минимизировав суммарную длину всех построенных дорог.

### Формат входного файла

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 5\,000$ ) — количество городов в Байтландии. Каждая из следующих  $n$  строк содержит по два целых числа  $x_i, y_i$  — координаты  $i$ -го города ( $-10\,000 \leq x_i, y_i \leq 10\,000$ ). Никакие два города не расположены в одной точке.

### Формат выходного файла

Первая строка выходного файла должна содержать минимальную суммарную длину дорог. Выведите число с точностью не менее  $10^{-3}$ .

### Пример

unionday.in	unionday.out
6 1 1 7 1 2 2 6 2 1 3 7 3	9.6568542495