

## Distance. Расстояние между вершинами

Имя входного файла: `distance.in`  
Имя выходного файла: `distance.out`

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

*Спектакль преподавателей ЛКШ. июль-2007*

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

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно. 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 ребер по одному на строке. Ребро номер  $i$  описывается тремя натуральными числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n, 0 \leq w_i \leq 100\,000$ ).

$n \leq 5\,000, m \leq 100\,000$ .

### Формат выходного файла

Первая строка выходного файла должна содержать одно натуральное число — вес минимального пути между вершинами  $s$  и  $t$ .

Если путь из  $s$  в  $t$  не существует, выведите -1.

### Пример

<code>distance.in</code>	<code>distance.out</code>
4 4	3
1 3	
1 2 1	
3 4 5	
3 2 2	
4 1 4	

## Multiple (\*). Наименьшее кратное

Имя входного файла: `multiple.in`  
Имя выходного файла: `multiple.out`

Дано число  $X$  и множество цифр  $D$ . Требуется дописать к  $X$  минимальное количество цифр из  $D$ , чтобы получившееся число делилось на  $k$ . При этом получившееся число должно быть минимально возможным.

Каждую цифру из множества можно использовать несколько раз.

### Формат входного файла

Первая строка входного файла содержит два натуральных числа  $X$  и  $k$  ( $1 \leq X \leq 10^{1000}, 2 \leq k \leq 100\,000$ ). Во второй строке записано количество цифр во множестве  $D$ . В третьей строке через пробел записаны эти цифры.

### Формат выходного файла

Единственная строка должна содержать минимальное число, полученное из  $X$  дописыванием цифр из  $D$  и кратное  $k$ . Если такого числа не существует, выведите -1.

### Пример

<code>multiple.in</code>	<code>multiple.out</code>
102 101 3 1 0 3	10201
202 101 3 1 0 3	202

## Island. Островные государства

Имя входного файла: `island.in`

Имя выходного файла: `island.out`

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.

Программист Вася живет в городе номер 1. С наступлением лета он собирается съездить в город  $N$  на Всебайтландское сборище 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 записано 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 посещенных городов, а во вторую выведите эти города в порядке посещения. Если минимальных путей несколько, выведите любой.

### Пример

island.in	island.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	1 3 1 4 5