

## Содержание

<b>11.Base [1/1]</b>	<b>3</b>
Задача 11A. Путь в случайном графе [0.5 sec, 256 mb]	3
<b>11.Advanced [2/4]</b>	<b>4</b>
Задача 11B. Path. Кратчайший путь [0.5 sec, 256 mb]	4
Задача 11C. Лабиринт знаний [0.5 sec, 256 mb]	5
Задача 11D. Цикл отрицательного веса [0.5 sec, 256 mb]	6
Задача 11E. Отрицательный цикл [0.5 sec, 256 mb]	7
<b>11.Hard [0/2]</b>	<b>8</b>
Задача 11F. Цикл минимального среднего веса [0.5 sec, 256 mb]	8
Задача 11G. Потенциал [0.5 sec, 256 mb]	9

### Общая информация:

Вход в констест: <http://contest.yandex.ru/contest/1892/>

Дедлайн на задачи: 10 дней, до 28-го ноября 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <https://compscicenter.ru/courses/algorithms-1/2015-autumn/>

Семинары ведут Сергей Копелиович ([burunduk30@gmail.com](mailto:burunduk30@gmail.com), [vk.com/burunduk1](https://vk.com/burunduk1)) и Глеб Леонов ([gleb.leonov@gmail.com](mailto:gleb.leonov@gmail.com), [vk.com/id1509292](https://vk.com/id1509292))

В каждом условии указан таймлимит для C/C++.

Таймлимит для Java примерно в 2-3 раза больше.

Таймлимит для Python примерно в 5 раз больше.

### C++:

Быстрый ввод-вывод.

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) переопределение стандартного аллокатора ускорит вашу программу:

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

### Java:

Быстрый ввод-вывод.

[http://acm.math.spbu.ru/~sk1/algo/input-output/java/java\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/java/java_common.html)

## 11.Base [1/1]

### Задача 11А. Путь в случайном графе [0.5 сек, 256 mb]

Дан случайный взвешенный ориентированный граф из  $n$  вершин,  $m$  рёбер. Гарантируется, что в графе нет отрицательных циклов. Ваша задача – найти расстояние от вершины  $s$  до вершины  $t$ . Что такое случайный граф? Рёбра равномерно случайного веса между равномерно случайными вершинами.

#### Формат входных данных

На первой строке  $n\ m\ s\ t$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 200\,000$ ,  $1 \leq s, t \leq n$ ). На следующих  $m$  строках тройки чисел  $a_i\ b_i\ w_i$  ( $1 \leq a, b \leq n$ ,  $|w| < 10^9$ ), рёбра из  $a_i$  в  $b_i$  веса  $w_i$ .

#### Формат выходных данных

Одно число – длину кратчайшего пути.

Если пути из  $s$  в  $t$  не существует, выведите число  $10^{18}$ .

#### Примеры

fb.in	fb.out
2 0 1 2	10000000000000000000
3 4 1 3 1 2 20 2 3 -15 1 3 10 3 1 -4	5

#### Замечание

Это простая задача. Достаточно реализовать алгоритм Форд-Беллмана с очередью.

## 11.Advanced [2/4]

### Задача 11В. Path. Кратчайший путь [0.5 sec, 256 mb]

Дан взвешенный ориентированный граф и вершина  $s$  в нем. Требуется для каждой вершины  $u$  найти длину кратчайшего пути из  $s$  в  $u$ .

#### Формат входных данных

Первая строка входного файла содержит  $n$ ,  $m$  и  $s$  — количество вершин, ребер и номер выделенной вершины соответственно ( $2 \leq n \leq 2\,000$ ,  $1 \leq m \leq 6\,000$ ).

Следующие  $m$  строк содержат описание ребер. Каждое ребро задается стартовой вершиной, конечной вершиной и весом ребра. Вес каждого ребра — целое число, не превосходящее  $10^{15}$  по модулю. В графе могут быть кратные ребра и петли.

#### Формат выходных данных

Выведите  $n$  строк — для каждой вершины  $u$  выведите длину кратчайшего пути из  $s$  в  $u$ , '\*' если не существует путь из  $s$  в  $u$  и '-' если не существует кратчайший путь из  $s$  в  $u$ .

#### Пример

path.in	path.out
6 7 1	0
1 2 10	10
2 3 5	-
1 3 100	-
3 5 7	-
5 4 10	*
4 3 -18	
6 1 -1	

#### Замечание

Задача на алгоритм Форда-Беллмана.

### Задача 11С. Лабиринт знаний [0.5 sec, 256 mb]

Участникам сборов подарили билеты на аттракцион “Лабиринт знаний”. Лабиринт представляет собой  $N$  комнат, занумерованных от 1 до  $N$ , между некоторыми из которых есть двери. Когда человек проходит через дверь, показатель его знаний изменяется на определенную величину, фиксированную для данной двери. Вход в лабиринт находится в комнате 1, выход — в комнате  $N$ . Каждый участник сборов проходит лабиринт ровно один раз и набирает некоторое количество знаний (при входе в лабиринт этот показатель равен нулю). Ваша задача — показать наилучший результат.

#### Формат входных данных

Первая строка входного файла содержит целые числа  $N$  ( $1 \leq N \leq 2000$ ) — количество комнат и  $M$  ( $0 \leq M \leq 10000$ ) — количество дверей. В каждой из следующих  $M$  строк содержится описание двери — номера комнат, из которой она ведет и в которую она ведет (через дверь в лабиринте можно ходить только в одну сторону), а также целое число, которое прибавляется к количеству знаний при прохождении через дверь (это число по модулю не превышает 10 000). Двери могут вести из комнаты в нее саму, между двумя комнатами может быть более одной двери.

#### Формат выходных данных

В выходной файл выведите “:)” — если можно пройти лабиринт и получить неограниченно большой запас знаний, “:(” — если лабиринт пройти нельзя, и максимальное количество набранных знаний в противном случае.

#### Пример

maze.in	maze.out
2 2 1 2 5 1 2 -5	5

#### Замечание

Задача на алгоритм Форда-Беллмана.

### Задача 11D. Цикл отрицательного веса [0.5 сек, 256 mb]

Дан ориентированный граф. Определите, есть ли в нем цикл отрицательного веса, и если да, то выведите его.

#### Формат входных данных

Во входном файле в первой строке число  $N$  ( $1 \leq N \leq 100$ ) — количество вершин графа. В следующих  $N$  строках находится по  $N$  чисел — матрица смежности графа. Все веса ребер не превышают по модулю 10 000. Если ребра нет, то соответствующее число равно 100 000.

#### Формат выходных данных

В первой строке выходного файла выведите «YES», если цикл существует или «NO» в противном случае. При его наличии выведите во второй строке количество вершин в искомом цикле и в третьей строке — вершины входящие в этот цикл в порядке обхода.

#### Пример

negcycle.in	negcycle.out
2	YES
0 -1	2
-1 0	1 2

#### Замечание

Можно решить эту задачу алгоритмом Флойда. А можно алгоритмом Форда-Беллмана.

### Задача 11Е. Отрицательный цикл [0.5 sec, 256 mb]

Дан взвешенный ориентированный граф. Требуется определить, содержит ли он цикл отрицательного веса. Гарантируется, что все вершины графа достижимы из первой.

#### Формат входных данных

Первая строка входного файла содержит два натуральных числа  $n$  и  $m$  — количество вершин и ребер графа соответственно ( $n \leq 1111$ ,  $m \leq 11111$ ). Следующие  $m$  строк содержат описание ребер по одному на строке. Ребро номер  $i$  описывается тремя числами  $b_i$ ,  $e_i$  и  $w_i$  — номера концов ребра и его вес соответственно ( $1 \leq b_i, e_i \leq n$ ,  $-100\,000 \leq w_i \leq 100\,000$ ). Обратите внимание, что в графе могут быть кратные ребра и петли.

#### Формат выходных данных

Первая строка выходного файла должна содержать **yes**, если граф содержит цикл отрицательного веса и **no** в противном случае.

#### Пример

negcycle.in	negcycle.out
4 4 2 1 -4 1 2 1 3 4 2 2 3 3	yes
4 6 2 1 4 1 2 1 3 4 2 2 3 3 1 1 2 1 2 2	no

#### Замечание

Задача на алгоритм Форда-Беллмана.

## 11.Hard [0/2]

### Задача 11F. Цикл минимального среднего веса [0.5 sec, 256 mb]

Дан ориентированный граф, содержащий  $n$  вершин и  $m$  ребер. Каждое ребро имеет некоторый вес. Простым циклом называется последовательность рёбер  $(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)$  такая, что все  $a_i$  различны,  $b_i = a_{i+1}$ ,  $b_k = a_1$ . Весом цикла называется сумма весов составляющих его ребер. Средним весом цикла называется отношение веса цикла к числу составляющих его ребер.

Требуется найти в графе **простой** цикл минимального среднего веса.

#### Формат входных данных

Первая строка входного файла содержит числа  $n$  и  $m$  ( $3 \leq n \leq 1000$ ,  $3 \leq m \leq 2000$ ). Следующие  $m$  строк описывают ребра графа. Каждое ребро описывается тремя целыми числами  $u_i$ ,  $v_i$  и  $w_i$ : номер начальной вершины, номер конечной вершины и вес ребра ( $1 \leq u_i, v_i \leq n$ ,  $-1000 \leq w_i \leq 1000$ ). Гарантируется, что в графе есть цикл. В графе может быть несколько ребер между одной и той же парой вершин, а также могут быть петли.

#### Формат выходных данных

На первой строке выведите число  $z$  — минимальный возможный средний вес цикла в заданном графе. На второй строке выведите число  $k$  — число ребер в таком цикле. На третьей строке выведите  $k$  чисел — номера ребер в порядке обхода вдоль цикла. Ребра пронумерованы от 1 до  $m$  в порядке, в котором они заданы во входном файле. Если возможных циклов несколько, можно вывести любой. Требуется вывести простой цикл. Число  $z$  должно быть выведено с абсолютной или относительной погрешностью  $10^{-9}$ .

#### Пример

meancycle.in	meancycle.out
5 8	2.5
1 2 10	4
2 3 1	2 3 4 7
3 4 2	
4 5 3	
5 1 8	
5 5 7	
5 2 4	
3 5 4	

#### Замечание

Подумайте про бинарный поиск по ответу...



### Задача 11G. Потенциал [0.5 сек, 256 mb]

Дан взвешенный ориентированный граф. Пусть у каждой вершины есть потенциал  $\Phi_i$ . Тогда к весу каждого ребра прибавляется потенциал начала и вычитается потенциал конца.

Требуется найти такие целые  $\Phi_i$ , чтобы веса у всех рёбер были одинаковыми.

#### Формат входных данных

В первой строке задано целое число  $t$  — количество тестовых случаев.

В первой строке каждого тестового случая заданы целые числа  $n$  и  $m$  ( $1 \leq n \leq 300\,000$ ,  $0 \leq m \leq 300\,000$ ) — количество вершин и рёбер в графе. В следующих  $m$  строках задано по три целых числа  $x_i$ ,  $y_i$  и  $w_i$  ( $1 \leq x_i, y_i \leq n$ ,  $-10^9 \leq w_i \leq 10^9$ ) — начало, конец и вес ребра. Гарантируется, что граф не содержит кратных рёбер и петель.

Также гарантируется, что сумма всех  $n$  и  $m$  по всем тестовым случаям не превосходит 600 000.

#### Формат выходных данных

Для каждого тестового случая выведите «YES», если существует целочисленное решение, и «NO» в противном случае.

Если ответ положительный, то в следующей строке выведите  $n$  целых чисел — потенциалы вершин. Все выведенные числа должны быть не больше  $10^{18}$  по абсолютной величине. Гарантируется, что если ответ существует, то существует и ответ, удовлетворяющий этому ограничению.

Если возможных ответов несколько, выведите любой из них.

#### Пример

potential.in	potential.out
2	YES
5 4	0 -1 1 2 181
1 2 -1	YES
2 3 2	0 0 0 0 -1
3 4 1	
4 5 179	
5 5	
1 2 1	
2 3 1	
3 4 1	
4 5 0	
5 1 2	