

Содержание

Обязательные задачи	2
Задача А. Расстояние между вершинами [1 sec, 256 mb]	2
Задача В. Остовное дерево 2 [1 sec, 256 mb]	3
Задача С. Island. Островные государства [0.2 sec, 256 mb]	4
Обычные задачи	5
Задача D. Unionday. День Объединения [1 sec, 256 mb]	5
Задача E. Path. Кратчайший путь [1 sec, 256 mb]	6
Задача F. Разрезание графа [1 sec, 256 mb]	7
Дополнительные задачи	8
Задача G. Путешествия по случайным городам [2 sec, 256 mb]	8
Задача H. Кратчайший путь [1.1 sec, 256 mb]	10
Задача I. MST случайных точек [4 sec, 256 mb]	11

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь [примерами](#).

В некоторых задачах большой ввод и вывод. Пользуйтесь [быстрым вводом-выводом](#).

Обратите внимание, что ввод-вывод во всех задачах стандартный.

Обязательные задачи

Задача А. Расстояние между вершинами [1 sec, 256 mb]

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 входных данных

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 ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100$).

$n \leq 100\,000$, $m \leq 200\,000$.

Формат выходных данных

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

stdin	stdout
4 4 1 3 1 2 1 3 4 5 3 2 2 4 1 4	3

Задача В. Остовное дерево 2 [1 сек, 256 mb]

Требуется найти в связном графе остовное дерево минимального веса.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно. 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 ($1 \leq b_i, e_i \leq n$, $0 \leq w_i \leq 100\,000$). $n \leq 20\,000$, $m \leq 100\,000$.

Граф является связным.

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число — вес минимального остовного дерева.

Примеры

stdin	stdout
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Задача C. Island. Островные государства [0.2 сек, 256 mb]

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 входных данных

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 выходных данных

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

stdin	stdout
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

Обычные задачи

Задача D. Unionday. День Объединения [1 sec, 256 mb]

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

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 5\,000$) — количество городов в Байтеландии. Каждая из следующих n строк содержит 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 выходных данных

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

Примеры

stdin	stdout
6 1 1 7 1 2 2 6 2 1 3 7 3	9.65685

Задача E. Path. Кратчайший путь [1 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 ($2 \leq n \leq 2\,000$, $1 \leq m \leq 5\,000$).

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

stdin	stdout
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	

Задача F. Разрезание графа [1 сек, 256 mb]

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 входных данных

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 -ая 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 выходных данных

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

stdin	stdout
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	

Дополнительные задачи

Задача G. Путешествия по случайным городам [2 сек, 256 mb]

Софи — путешественница, ей нравится ездить по случайным городам, исследовать местные достопримечательности, улицы, площади и всё такое. Для этого ей часто нужно знать, как добраться из одного места в другое как можно быстрее. Так как город, в котором она находится сейчас, довольно большой и довольно случайный, Софи очень сложно в нем ориентироваться, и ей нужна ваша помощь.

Карта города может быть представлена как неориентированный граф без петель и кратных рёбер, где вершины соответствуют интересным местам, а ребро между двумя такими местами означает, что Софи может добраться из одного в другое (или наоборот) за одну минуту.

Кроме того, город случаен в следующем смысле: его граф состоит из ровно N вершин и M рёбер, и выбирается случайно и равномерно среди всех таких графов.

Софи нужно получить ответы на Q запросов вида «какова длина кратчайшего пути между вершинами u и v ?» Пара вершин в каждом запросе выбирается случайно и равномерно среди всех пар *различных* вершин в графе.

Дополнительно, во всех тестах $N = 10^5$, $M = 3 \cdot 10^5$, $Q = 10^4$. Это условие и условие на случайность не распространяются на тест из условия, однако ваше решение должно проходить и его.

Формат входных данных

В первой строке входа два целых числа N и M ($N = 10^5$, $M = 3 \cdot 10^5$), разделённые пробелом.

В следующих M строках описания рёбер: в каждой строке по два целых числа, разделённых пробелом — номера концов соответствующего ребра, вершины нумеруются от 1 до N . Петель и кратных рёбер нет.

Следующая строка содержит одно целое число Q ($Q = 10^4$) — количество запросов.

В следующих M строках описания запросов: в каждой строке по два целых числа, разделённых пробелом — номера вершин, минимальное расстояние между которыми нужно найти.

Формат выходных данных

Выведите Q строк, в каждой ответ на очередной запрос: длину кратчайшего пути между данными вершинами, либо «-1» (без кавычек), если пути между ними не существует.

Пример

stdin	stdout
6 5	1
1 2	2
2 3	3
1 3	2
1 4	-1
4 5	
5	
1 3	
4 2	
3 5	
5 1	
4 6	

Подсказка по решению

Не один bfs, а целых два!

Задача Н. Кратчайший путь [1.1 sec, 256 mb]

Надеюсь, все вы умеете искать в ориентированном графе кратчайший путь. В этой задаче вам предлагается свое умение продемонстрировать.

Вам дан ориентированный взвешенный граф. Веса ребер — целые числа от 1000 до 2000. Нужно несколько раз (не более 1000) ответить на следующий запрос: длина кратчайшего пути из некоторой вершины s в некоторую вершину t .

Формат входных данных

На первой строке числа N и M ($1 \leq N \leq 25\,000$, $0 \leq M \leq 50\,000$) — количество вершин и ребер нашего графа, соответственно. Вершины нумеруются целыми числами от 1 до N . Далее M строк содержат информацию о ребрах графа. Каждое ребро задается тремя числами — номер начала, номер конца и вес. Все веса — целые числа от 1000 до 2000. В графе могут быть и петли, и кратные ребра. Следующая строка содержит число K ($1 \leq K \leq 1000$) — количество запросов. В следующих K строках задаются запросы. Каждый запрос описывается двумя числами — из какой вершины, и в какую должен вести путь.

Формат выходных данных

Для каждого запроса выведите на отдельной строке целое число — длину кратчайшего пути. Если кратчайшего пути не существует следует вывести -1 .

Пример

stdin	stdout
5 5	-1
1 2 2000	3000
1 3 1000	0
1 4 1200	2000
2 3 1500	
3 4 1500	
4	
1 5	
2 4	
3 3	
1 2	

Замечание

Путем в графе называется такая последовательность ребер, что конец i -го совпадает с началом $i+1$ -го. Длиной пути называется суммарный вес ребер. Путь является кратчайшим, если его длина минимальна.

Подсказка по решению

Вы можете потолкать Дейкстру или Форд-Беллмана, но есть решение лучше.

Задача I. MST случайных точек [4 sec, 256 mb]

Даны n различных точек на плоскости. Координаты точек — целые числа от 0 до 30 000 включительно. Точки выбраны *случайно* в следующем смысле: рассмотрим все возможные наборы из n различных точек на плоскости с заданными ограничениями на координаты и выберем из них случайно и равновероятно один набор.

Вы можете провести отрезок между любыми двумя заданными точками. Длина отрезка между точками с координатами (x_1, y_1) и (x_2, y_2) равна $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Будем говорить, что точки a и b *связаны*, если они соединены отрезком, или же существует точка d , которая связана и с a , и с b . Ваша задача — провести отрезки минимальной суммарной длины так, чтобы все точки были связаны.

Формат входных данных

В первой строке ввода задано целое число n ($2 \leq n \leq 50\,000$). Следующие n строк содержат координаты точек. Гарантируется, что все точки различны. Кроме того, во всех тестах, кроме примера, гарантируется, что точки выбраны случайно, как описано в условии.

Формат выходных данных

В первой строке выведите вещественное число w — суммарную длину отрезков. В следующих $(n - 1)$ строках выведите отрезки, по одному на строке. Каждый отрезок следует выводить как два числа от 1 до n , обозначающие номера точек, являющихся концами этого отрезка.

Пусть на самом деле суммарная длина выведенных вами отрезков равна w^* , а суммарная длина отрезков в оптимальном ответе равна w_{opt} . Тогда ваш ответ будет считаться верным, если

$$\max \left(\left| \frac{w}{w^*} - 1 \right|, \left| \frac{w^*}{w_{\text{opt}}} - 1 \right| \right) < 10^{-12}.$$

Пример

stdin	stdout
4	22.02362358924615
0 10	1 2
5 6	2 3
10 0	4 2
0 0	