

Содержание

Must have	2
Задача 8А. Разрезание графа [0.2 sec, 256 mb]	2
Задача 8В. Остовное дерево 2 [0.2 sec, 256 mb]	3
Обязательные задачи	4
Задача 8С. Unionday. День Объединения [0.3 sec, 256 mb]	4
Задача 8D. Ребра добавляются, граф растет [0.5 sec, 256 mb]	5
Дополнительные задачи	6
Задача 8Е. Два китайца и дерево [0.2 sec, 256 mb]	6
Задача 8F. Возьми себе за правило: летай всегда GraphAero! [0.5 sec, 256 mb]	7
Задача 8G. MST случайных точек [2 sec, 256 mb]	9
Задача 8H. Таможенные пошлины [0.5 sec, 256 mb]	10

Вы не умеете читать/выводить данные, открывать файлы? Воспользуйтесь **примерами**.

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

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

Обратите внимание на компилятор GNU C++11 5.1.0 (TDM-GCC-64) inc, который позволяет пользоваться **дополнительной библиотекой**. Под ним можно сдать **вот это**.

Must have

Задача 8А. Разрезание графа [0.2 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	

Замечание

Это простая задача на DSU.

Задача 8В. Остовное дерево 2 [0.2 sec, 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

Замечание

Можно написать Прима с кучей или Краскала...

Обязательные задачи

Задача 8С. Unionday. День Объединения [0.3 сек, 256 mb]

В Байтландии 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 Байтландии 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 Байтландии. 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 8D. Ребра добавляются, граф растёт [0.5 сек, 256 mb]

В неориентированный граф последовательно добавляются новые ребра. Изначально граф пустой. После каждого добавления нужно говорить, является ли текущий граф двудольным.

Формат входных данных

На первой строке n — количество вершин, m — количество операций «добавить ребро». Следующие m строк содержат пары чисел от 1 до n — описание добавляемых ребер.

$$1 \leq n, m \leq 300\,000$$

Формат выходных данных

Выведите в строчку m нулей и единиц. i -й символ должен быть равен единице, если граф, состоящий из первых i ребер, является двудольным.

Примеры

stdin	stdout
3 3 1 2 2 3 3 1	110

Замечание

Решение с бинарным поиском скорее всего не зайдёт по времени. Используйте DSU.

Дополнительные задачи

Задача 8E. Два китайца и дерево [0.2 сек, 256 mb]

Дан ориентированный взвешенный граф. Выбрать множество рёбер минимально возможного суммарного веса, чтобы все вершины были достижимы из первой. Гарантируется, что такое множество существует.

Формат входных данных

Входные данные содержат описание одного или более тестов.

На первой строке количества вершин и рёбер в графе n ($1 \leq n \leq 200$) и m ($1 \leq m \leq 100\,000$). На следующих m строках рёбра в формате " $a_i b_i w_i$ " ($1 \leq a_i, b_i \leq n$, $-10^5 < w_i < 10^5$), что обозначает ребро из a_i в b_i веса w_i . В графе могут быть петли и кратные рёбра.

Сумма n по всем тестам не более 200. Сумма m по всем тестам не более 100 000.

Формат выходных данных

Для каждого теста выведите две строки ответа.

На первой строке выведите суммарный вес выбранных рёбер и их количество. Во второй строке выведите номера выбранных рёбер, числа от 1 до m в произвольном порядке.

Примеры

stdin	stdout
3 3	-14 3
1 2 -3	1 2 3
2 3 -1	21 3
3 1 -10	1 4 3
4 5	3 5
1 2 10	1 2 3 4 6
2 4 2	
1 3 10	
3 4 1	
1 4 10	
5 6	
1 2 1	
2 3 1	
3 4 1	
4 5 1	
1 5 2	
5 3 -1	

Задача 8F. Возьми себе за правило: летай всегда GraphAero! [0.5 sec, 256 mb]

Наконец авиаперевозки стали доступны всем и каждому! Однако, из-за жёсткой конкуренции в сфере пассажироперевозок осталось только две авиакомпании: «GraphAero Airlines» и «Aerofloat».

Авиакомпания «GraphAero Airlines» активно развивается. Ведь для получения большей прибыли... простите, для удобства пассажиров каждый месяц компания добавляет один новый рейс. Компании «Aerofloat» остаётся довольствоваться тем, что остаётся. А именно, единственная возможность удержаться на плаву — добавлять рейсы, дублирующие самые загруженные рейсы компании «GraphAero Airlines». Рейс является самым загруженным, если существует такая пара городов, что можно долететь (возможно, с пересадками) из одного города в другой, используя рейсы авиакомпании, но если этот рейс отменить — то долететь будет невозможно. Аналитикам компании «Aerofloat» необходимо постоянно контролировать ситуацию — сколько в данный момент существует самых загруженных рейсов.

Поскольку вы уже давно мечтаете летать по льготным ценам (скидка $10^{-5}\%$), вы решили оказать посильную помощь. Помните: самолёты летают по всему миру! Между двумя крупными городами может быть более одного рейса, а города бывают настолько большими, что самолёты могут летать в пределах одного города. Рейсами можно пользоваться как в одну, так и в другую сторону.

Формат входных данных

Первая строка входного файла содержит целое число N ($1 \leq N \leq 100\,000$) — количество городов и M ($0 \leq M \leq 100\,000$) — изначальное число рейсов компании «GraphAero Airlines». Далее следует M строк, в каждой содержится описание очередного рейса — номера двух городов, между которыми осуществляется рейс. В следующей строке содержится число K ($1 \leq K \leq 100\,000$) — количество добавленных рейсов. Далее содержится описание добавленных рейсов в таком же формате.

Формат выходных данных

После каждого добавления нового рейса выведите на отдельной строке одно число — количество самых загруженных рейсов.

Примеры

stdin	stdout
4 0	1
4	2
1 2	3
2 3	0
3 4	
1 4	
4 3	3
1 2	2
2 3	1
3 4	0
4	
1 1	
1 2	
1 3	
1 4	

Замечание

Это несложная задача на тему DSU. Её можно решать в offline. Хранить скорее всего придётся целых два DSU...

Задача 8G. MST случайных точек [2 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.

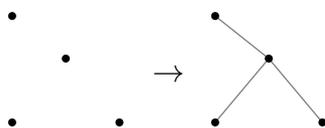
Пусть на самом деле суммарная длина выведенных вами отрезков равна w^* , а суммарная длина отрезков в оптимальном ответе равна w_{opt} . Тогда ваш ответ будет считаться верным, если

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

Пример

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

Иллюстрация



Замечание

Вы уже знаете из последней теорпрактики, каким алгоритмом воспользоваться. Вопрос в том, как найти ближайшие $\mathcal{O}(1)$ точек? Воспользуйтесь методом “проекция на случайную прямую”. Приветствуются попытки решить задачу любым другим способом =)

Задача 8N. Таможенные пошлины [0.5 sec, 256 mb]

Недавно королева страны AlgoLand придумала новый способ отмывания денег для своего королевского двора. Она решила, что всякий житель, желающий совершить путешествие из одного города страны в другой, должен расплатиться за это желание своими деньгами.

В стране AlgoLand есть N городов, пронумерованных от 1 до N . Некоторые города соединены дорогами, движение по которым разрешено в двух направлениях. Начиная движение по какой-нибудь дороге, путешественник обязательно должен доехать до ее конца.

Предположим теперь, что житель страны хочет совершить путешествие из города A в город B . Новый указ королевы гласит, что при проезде по любой дороге страны во время этого путешествия, полицейские могут взять с этого жителя таможенную пошлину в пользу королевского двора (а могут и не взять). Если при этом у жителя недостаточно денег для уплаты пошлины, то он автоматически попадает в тюрьму. Указ также устанавливает величину пошлины для каждой дороги страны. Так как королева заботится о жителях своей страны, то она запретила полицейским брать с жителя пошлину более чем три раза во время одного путешествия.

Отметим, что если существует несколько способов попасть из города A в город B , то житель может выбрать для путешествия любой из них по собственному желанию.

Напишите программу, которая определяет, какую минимальную сумму денег должен взять с собой житель, чтобы гарантированно не попасть в тюрьму во время путешествия.

Формат входных данных

Первая строка входного файла содержит числа N и M ($2 \leq N \leq 10\,000$, $1 \leq M \leq 100\,000$), разделенные пробелом — количества городов и дорог. Следующие M строк описывают дороги. Каждая из этих строк описывает одну дорогу и содержит три числа X , Y , Z ($1 \leq X, Y \leq N$; $X \neq Y$; $1 \leq Z \leq 1\,000\,000\,000$), разделенных пробелами, означающие, что дорога соединяет города X и Y и пошлина за проезд по ней равна Z денежных единиц. Все числа Z целые. Последняя строка содержит числа A и B ($1 \leq A, B \leq N$; $A \neq B$) — номера начального и конечного городов путешествия. Гарантируется, что существует хотя бы один способ проезда из A в B .

Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное минимальной сумме денег, которую должен взять с собой житель, чтобы иметь возможность совершить путешествие из города A в город B и при этом гарантированно не попасть в тюрьму независимо от действий полицейских.

Пример

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