Содержание

Общая информация:

Вход в контест: `http://contest.yandex.ru/contest/1917/`
Дедлайн на задачи: 10 дней, до 5-го декабря 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, vk.com/burunduk1) и Глеб Леонов (gleb.leonov@gmail.com, vk.com/id1509292)

В каждом условии указан таймлимит для `C/C++`.
Таймлиминт для `Java` примерно в 2-3 раза больше.
Таймлиминт для `Python` примерно в 5 раз больше.

C++:

Быстрый ввод-вывод.
`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`

12.Base [1/1]

Задача 12A. Соединение точек [0.1 sec, 256 mb]

Даны N точек на плоскости. Требуется провести отрезки между некоторыми парами точек таким образом, чтобы, во-первых, из любой данной точки в любую можно было пройти по этим отрезкам, а во-вторых, суммарная длина проведённых отрезков была минимальна.

Формат входных данных

В первой строке входного файла задано число N — количество точек ($1 \leqslant N \leqslant 200$). Следующие N строк содержат по два числа X_i Y_i каждая через пробел — координаты i-ой точки ($-1000 \leqslant X_i, Y_i \leqslant 1000$). Никакие две данные точки не совпадают, никакие три не лежат на одной прямой. Все числа во входном файле целые.

Формат выходных данных

В первой строке выходного файла выведите L — суммарную длину проведённых отрезков с точностью не менее шести десятичных знаков после запятой. Во второй строке выведите K — их количество. В следующих K строках выведите по два числа A_j B_j через пробел в каждой — номера точек, соединённых j-ым отрезком ($1 \leqslant A_j, B_j \leqslant N$, $A_j \neq B_j$). Точки нумеруются с единицы в том порядке, в котором они даны во входном файле. Если ответов с минимальным L несколько, разрешается выводить любой из них.

Примеры

connect.in	connect.out
4 0 0 0 1 1 0 1 1	3 3 1 2 2 4 4 3
5 0 0 0 2 1 1 3 0 3 2	7.064495 4 3 1 3 2 3 4 4 5

12.Advanced [3/5]

Задача 12B. Разрезание графа [0.3 sec, 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 ($1 \leqslant n \leqslant 50\,000$, $0 \leqslant m \leqslant 100\,000$, $m \leqslant k \leqslant 150\,000$).

Следующие m строк задают рёбра графа; i-ая из этих строк содержит два числа u_i и v_i ($1 \leqslant u_i, v_i \leqslant n$), 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" ($1 \leqslant u, v \leqslant n$), которая означает, что из графа удаляют ребро между вершинами u и v. Операция типа `ask` задаётся строкой "`ask` u v" ($1 \leqslant u, v \leqslant n$), 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

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

Задача 12C. Unionday. День Объединения [1 sec, 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 ($1 \leqslant n \leqslant 5\,000$) — количество городов в Байтландии. Каждая из следующих n строк содержит два целых числа x_i, y_i — координаты i-го города ($-10\,000 \leqslant x_i, y_i \leqslant 10\,000$). 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

unionday.in	unionday.out
6 1 1 7 1 2 2 6 2 1 3 7 3	9.65685

Задача 12D. Болото [0.1 sec, 256 mb]

Иван-Царевич хочет спасти из плена Василису Прекрасную. По пути к темнице, где Кощей Бессмертный держит пленницу, есть болото с параллельными бесконечно длинными берегами ширины H. В болоте имеется N кочек, i –я кочка имеет координаты x_i, y_i. Ось OX направлена параллельно берегу болота, а ось OY направлена перпендикулярно берегу болота от начального берега к конечному, точки начального берега имеют координату $Y = 0$.

Требуется определить, какой минимальной длиной прыжка должен обладать Иван –Царевич, чтобы перебраться через болото.

Формат входных данных

Во входном файле в первой строке находятся числа H ($1 \leqslant H \leqslant 30\,000$) и N ($1 \leqslant N \leqslant 100$). В следующих N строках записаны координаты кочек x_i, y_i ($1 \leqslant x_i, y_i \leqslant 30\,000$). Число H и все координаты — целые числа.

Формат выходных данных

В выходной файл нужно вывести единственное число — минимальную длину прыжка с точностью до 6 знаков после точки.

swamp.in	swamp.out
10 3 1 3 3 7 6 6	4.472135955

Задача 12E. Остовное дерево 2 [0.25 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 ($1 \leqslant b_i, e_i \leqslant n$, $0 \leqslant w_i \leqslant 100\,000$). $n \leqslant 20\,000, m \leqslant 100\,000$.

Граф является связным.

Формат выходных данных

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

Примеры

spantree2.in	spantree2.out
4 4 1 2 1 2 3 2 3 4 5 4 1 4	7

Задача 12F. Ребра добавляются, граф растет [0.5 sec, 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.

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.

Система оценки

Подзадача 1 (25 баллов) $1 \leqslant n, m \leqslant 1\,000$.
Подзадача 2 (50 баллов) $1 \leqslant n, m \leqslant 50\,000$.
Подзадача 3 (25 баллов) $1 \leqslant n, m \leqslant 300\,000$.

Примеры

addedge.in	addedge.out
3 3 1 2 2 3 3 1	110

12.Hard [0/3]

Задача 12G. MST случайных точек [2.5 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 ($2 \leqslant n \leqslant 50\,000$). 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.

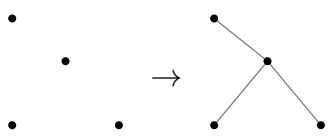
Пусть на самом деле суммарная длина выведенных вами отрезков равна w^*, а суммарная длина отрезков в оптимальном ответе равна w_{opt}. Тогда ваш ответ будет считаться верным, если

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

Пример

randommst.in	randommst.out
4 0 10 5 6 10 0 0 0	22.02362358924615 1 2 2 3 4 2

Иллюстрация



Задача 12H. Чётность [0.1 sec, 256 mb]

Вы играете со своим другом в следующую игру. Ваш друг записывает последовательность, состоящую из нулей и единиц. Вы выбираете непрерывную подпоследовательность (например, подпоследовательность от третьей до пятой цифры включительно) и спрашиваете его, чётное или нечётное количество единиц содержит эта подпоследовательность. Ваш друг отвечает, после чего вы можете спросить про другую подпоследовательность, и так далее. Ваша задача – угадать всю последовательность чисел. Но вы подозреваете, что некоторые из ответов вашего друга могут быть неверными, и хотите уличить его в обмане. Вы решили написать программу, которая получит наборы ваших вопросов вместе с ответами друга и найдет первый ответ, который гарантированно неверен. Это должен быть такой ответ, что существует последовательность, удовлетворяющая ответам на предыдущие вопросы, но никакая последовательность не удовлетворяет этому ответу.

Формат входных данных

Ввод содержит несколько тестов. Первая строка каждого теста содержит одно число, равное длине последовательности нулей и единиц. Эта длина не превосходит 10^9. Во второй строке находится одно неотрицательное целое число – количество заданных вопросов и ответов на них. Количество вопросов и ответов не превышает 5 000. Остальные строки содержат вопросы и ответы. Каждая строка содержит один вопрос и ответ на этот вопрос: два целых числа (позиции первой и последней цифр выбранной подпоследовательности) и одно слово – “even” или “odd” – ответ, сообщающий чётность количества единиц в выбранной подпоследовательности, где “even” означает чётное количество единиц, а “odd” означает нечётное количество. Ввод заканчивается строкой, содержащей -1.

Формат выходных данных

Каждая строка вывода должна содержать одно целое число X. Число X показывает, что существует последовательность нулей и единиц, удовлетворяющая первым X условиям чётности, но не существует последовательности, удовлетворяющей $X+1$ условию. Если существует последовательность нулей и единиц, удовлетворяющая всем заданным условиям, то число X должно быть равно количеству всех заданных вопросов.

Примеры

parity.in	parity.out
10 5 1 2 even 3 4 odd 5 6 even 1 6 even 7 10 odd -1	3

Задача 12I. Возьми себе за правило — летай всегда GraphAero! [0.4 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 ($1 \leqslant N \leqslant 100\,000$) — количество городов и M ($0 \leqslant M \leqslant 100\,000$) — 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 ($1 \leqslant K \leqslant 100\,000$) — 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

bridges.in	bridges.out
4 0 4 1 2 2 3 3 4 1 4	1 2 3 0
4 3 1 2 2 3 3 4 4 1 1 1 2 1 3 1 4	3 2 1 0