Содержание

Общая информация:

Вход в контест: http://contest.yandex.ru/contest/3142/
Дедлайн на задачи: 9 дней, до 2016-11-12 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-1/2016-autumn/
Семинары ведут Сергей Копелиович (burunduk30@gmail.com, vk.com/burunduk1) и Алексей Кладов (aleksey.kladov@gmail.com).

В каждом условии указан таймлимит для `C/C++`.
Таймлиминт для `Java` примерно в 2-3 раза больше.
Таймлиминт для `Python` примерно в 6 раз больше.

C++:

Быстрый ввод-вывод.
http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html Более подробно про 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

9.base [2/2]

Задача 9A. Расстояние от корня [0.1 sec, 256 mb]

В заданном корневом дереве найдите вершины, максимально удалённые от корня. Расстоянием между вершинами считается количество рёбер в пути.

Формат входных данных

В первой строке задано n — количество вершин в дереве ($1 \leqslant n \leqslant 100$). В следующих $n-1$ строках заданы вершины, являющиеся предками вершин 2, 3, ..., n. Вершина 1 является корнем дерева.

Формат выходных данных

В первой строке выведите максимальное расстояние от корня до остальных вершин дерева. Во второй строке выведите, сколько вершин дерева находятся от корня на таком расстоянии. В третьей строке выведите номера этих вершин через пробел в порядке возрастания.

Примеры

rootdist.in	rootdist.out
3 1 1	1 2 2 3
3 1 2	2 1 3

Задача 9B. Bridges. Мосты [0.2 sec, 256 mb]

Дан неориентированный граф. Требуется найти все мосты в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leqslant 20\,000$, $m \leqslant 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество мостов в заданном графе. На следующей строке выведите b целых чисел — номера ребер, которые являются мостами, в возрастающем порядке. Ребра нумеруются с единицы в том порядке, в котором они заданы во входном файле.

Пример

bridges.in	bridges.out
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6	1 3

9.advanced [3/5]

Задача 9C. Мосты и компоненты [0.2 sec, 256 mb]

Дан неориентированный граф (не обязательно связный). Граф может содержать петли и кратные ребра.

Выведите все компоненты реберной двусвязности графа (максимальные подмножества вершин, такие что подграф на них не теряет связность при удалении любого ребра).

Формат входных данных

Первая строка содержит числа n и m ($1 \leqslant n \leqslant 100\,000$, $0 \leqslant m \leqslant 100\,000$) — количество вершин и ребер в графе.

Следующие m строк задают ребра графа.

Формат выходных данных

В первой строке выведите количество компонент, в следующих за ней строках выведите сами компоненты, по одной на строку.

Вершины в каждой компоненте должны идти в возрастающем порядке, компоненты нужно вывести в лексикографическом порядке.

Компонента – вектор номеров своих вершин. Лексикографически сравниваются вектора.

Примеры

bridges.in	bridges.out
3 2 1 2 2 3	3 1 2 3
3 3 1 2 2 3 3 1	1 1 2 3
2 2 1 2 1 2	1 1 2
7 8 1 5 5 6 1 6 5 4 4 3 4 2 3 2 7 2	3 1 5 6 2 3 4 7

Задача 9D. Points. Точки сочленения [0.2 sec, 256 mb]

Дан неориентированный граф без петель а кратных рёбер. Требуется найти все точки сочленения в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количество вершин и ребер графа соответственно ($n \leqslant 20\,000$, $m \leqslant 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leqslant b_i, e_i \leqslant n$).

Формат выходных данных

Первая строка выходного файла должна содержать одно натуральное число b — количество точек сочленения в заданном графе. На следующей строке выведите b целых чисел — номера вершин, которые являются точками сочленения, в возрастающем порядке.

Пример

points.in	points.out
9 12 1 2 2 3 4 5 2 6 2 7 8 9 1 3 1 4 1 5 6 7 3 8 3 9	3 1 2 3

Задача 9E. Из истории банка Гринготтс [0.1 sec, 256 mb]

Чтобы понять название задачи, можно прочитать красивую легенду.
http://acm.timus.ru/problem.aspx?space=1&num=1441

Задача же заключается в том, чтобы рёбра неориентированного графа разбить на минимальное число путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leqslant n \leqslant 20\,000$) и число рёбер m ($1 \leqslant m \leqslant 20\,000$). Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i b_i ($1 \leqslant a_i, b_i \leqslant n$). Между каждыми двумя вершинами не более одного ребра. Граф связен.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей описание очередного пути – номера вершин в порядке прохождения.

Примеры

euler.in	euler.out
7 7 1 2 4 1 6 7 5 7 7 4 2 3 4 2	3 5 7 4 2 1 4 2 3 6 7

Задача 9F. Зависимости между функциями [0.1 sec, 256 mb]

Даны названия функций и зависимости между ними. Нужно выписать названия функций по одному разу в таком порядке, что:

- Если функция A зависит от функции B, функция B должна быть выписана раньше функции A.
- Если предыдущий критерий позволяет в какой-то момент выписать более чем одну функцию, то выписана должна быть та из них, название которой лексикографически минимально.

Известно, что между функциями нет циклических зависимостей.

Формат входных данных

В первой строке входного файла задано целое число n — количество функций ($1 \leqslant n \leqslant 50$). Следующие n строк содержат описания функций. Каждая из этих строк имеет вид $\texttt{name}_i\ d_{i,1}\ d_{i,2}\ \ldots\ d_{i,k_i}$; здесь $\texttt{name}_i$ — имя функции, а $d_{i,l}$ — номера функций (считая с нуля), от которых зависит данная. Имя функции непусто и состоит из не более чем 50 заглавных букв латинского алфавита; имена всех функций различны. Соседние элементы строки отделены друг от друга одним пробелом. Длина каждой строки не превосходит 200 символов.

Формат выходных данных

Выведите в выходной файл n строк. В каждой строке выведите название одной из функций. Функции должны быть перечислены в требуемом порядке.

Примеры

functions.in	functions.out
3 B 1 C A 1	C A B
3 B 1 C A 0	C B A
10 K A B 1 1 C 2 D 3 E 4 F 5 G 6 H 7 I 8	A B C D E F G H I K

Задача 9G. Дорожные работы [3 sec, 256 mb]

В республике Икс издавна действует двухпартийная система. Каждый год граждане, имеющие избирательные права, голосуют, какой партии они больше доверяют — партии Мошенников или партии Грабителей, и в течение этого года вся реальная власть сосредоточена в руках избранной партии.

В последние M лет между партиями разразилась нешуточная война по перестройке дорожной сети республики «под себя». Партия Мошенников стремится построить как можно больше государственных дорог, чтобы прикарманить побольше бюджетных денег на их «обслуживание», а партия Грабителей стремится сделать платными как можно большее число дорог. Движение на всех дорогах республики Икс двустороннее.

Известно, что в течение одного года правления партии Мошенников удавалось построить ровно одну новую дорогу (которая поначалу является бесплатной), а партии Грабителей — ввести плату за проезд по одной из бесплатных на текущий момент дорог (при этом деньги на содержание этой дороги выделяются уже не из бюджета, а из средств, вырученных за проезд).

Президент республики, в настоящее время не имеющий реального политического влияния, решил привлечь внимание общественности к проблеме дорог. Он назвал дорожную сеть *удобной* (для простых граждан), если из любого города можно доехать до любого, используя только бесплатные дороги, но при этом количество бесплатных дорог (а, соответственно, и бюджетные средства на их содержание, полученные сбором налогов с граждан республики) — минимально возможное.

Вам поручено написать программу, которая определяет, была ли дорожная сеть удобной по завершении i-го года «дорожной войны».

Формат входных данных

В первой строке ввода заданы два числа — N ($1 \leqslant N \leqslant 1000$), число городов в республике Икс, и M ($1 \leqslant M \leqslant 100\,000$), продолжительность порядком затянувшейся «дорожной войны». Далее следуют M строк, первый символ каждой из которых — это `F`, если в данный год у власти была партия Мошенников, и `R` — если партия Грабителей, а далее в строке следуют два числа — номера городов u_i и v_i — пара городов, дорога между которыми стала объектом пристального внимания соответствующей партии (была построена новая дорога, если у власти была партия Мошенников, и одна из существующих дорог была сделана платной, если у власти была партия Грабителей). Вполне возможна ситуация, когда между двумя городами окажется более одной дороги, или будет построена дорога из города в себя — мало ли, что там удумает партия Мошенников.

Гарантируется, что входные данные корректны, то есть, все числа u_i и v_i лежат в пределах от 1 до N, и если известно, что в какой-то год дорога между двумя городами была сделана платной, то это значит, что перед началом года была хотя бы одна бесплатная дорога между этими городами.

Формат выходных данных

Для каждого года выведите в отдельной строке `YES`, если дорожная сеть по завершении соответствующего года была удобной, и `NO` в противном случае.

Пример

roadwork.in	roadwork.out
4 8 F 1 2 F 1 3 R 1 3 F 2 3 F 3 4 F 1 3 R 1 3 F 1 1	NO NO NO NO YES NO YES NO

9.hard [0/4]

Задача 9H. King's Assassination [0.4 sec, 256 mb]

Дан граф из n вершин и m ребер. Граф ориентированный. Нужно определить число вершин, содержащихся на всех путях из s в t (сами s и t учитывать не нужно).

Формат входных данных

Первая строка содержит n, m, s и t ($2 \leqslant n \leqslant 100\,000$, $1 \leqslant m \leqslant 300\,000$, $1 \leqslant s, t \leqslant n$, $s \neq t$).

Следующие m строк содержат пары чисел x_i и y_i — индексы вершин от 1 до n. Это означает что есть дорога из вершины с номером x_i в вершину с номером y_i.

Формат выходных данных

Число вершин k. Далее k чисел — номера вершин в возрастающем порядке.

Примеры

assassination.in	assassination.out
4 3 1 4 1 2 2 3 3 4	2 2 3
4 4 1 4 1 2 2 3 3 4 1 3	1 3
4 5 1 4 1 2 2 3 3 4 1 3 2 4	0

Задача 9I. Раскраска в три цвета [0.1 sec, 256 mb]

Петя нарисовал на бумаге n кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

Формат входных данных

Первая строка содержит два целых числа n и m — количество кружков и количество линий, которые нарисовал Петя, соответственно ($1 \leqslant n \leqslant 1\,000$, $0 \leqslant m \leqslant 20\,000$).

Следующая строка содержит n символов из множества {'R', 'G', 'B'} — i-й из этих символов означает цвет, в который раскрашен i-й кружок ('R' — красный, 'G' — зеленый, 'B' — синий).

Следующие m строк содержат по два целых числа — пары кружков, соединенных отрезками.

Формат выходных данных

Выведите в выходной файл одну строку, состоящую из n символов из множества {'R', 'G', 'B'} — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово "Impossible".

Примеры

color.in	color.out
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	BBGR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

Задача 9J. Chip Installation [0.3 sec, 256 mb]

Новый ЧИП скоро установят в новый летательный апарат, недавно выпущенной компанией `Airtram`. ЧИП имеет форму диска. Есть n проводов, которые нужно подсоединить к ЧИПу.

Каждый провод можно подсоединить в один из двух разъемов, допустимых для этого провода. Все $2n$ разъемов расположены на границе диска. По кругу. Каждый провод имеет свой цвет. Для повышения безопасности два провода одного цвета не могут быть подсоединены к соседним разъемам.

Дана конфигурация разъемов на ЧИПе, найдите способ подсоединить все провода, не нарушающий условия про цвета.

Формат входных данных

Первая строка содержит число n — количество проводов ($1 \leqslant n \leqslant 50\,000$). Вторая строка содержит n целых чисел от 1 до 10^9 — цвета проводов. Цвета проводов могут совпадать. Третья строка содержит $2n$ целых чисел от 1 до n описывающих разъемы. Число обозначает номер провода, который может быть подсоединен к данному разъему. Каждое число от 1 до n встречается ровно дважды. Разъемы перечислены в порядке "по кругу". 1-й разъем является соседним со 2-м и так далее, не забудьте, что n-й является соседним с 1-м.

Формат выходных данных

Если не существует способа подключить все провода, выведите одно слово “`NO`”.

Иначе выведите “`YES`” и n целых чисел. Для каждого провода выведите номер разъема, к которому нужно подключить этот провод. Разъемы нумеруются числами от 1 до $2n$ в том порядке, в котором они даны во входном файле.

Примеры

chip.in	chip.out
2 1 1 1 1 2 2	YES 1 3
2 1 1 1 2 1 2	NO
2 1 2 1 2 1 2	YES 1 2

Задача 9K. Кодовый замок [0.4 sec, 256 mb]

Петя опоздал на тренировку по программированию! Поскольку тренировка проходит в воскресенье, главный вход в учебный корпус, где она проходит, оказался закрыт, а вахтёр ушёл куда-то по своим делам. К счастью, есть другой способ проникнуть в здание — открыть снаружи боковую дверь, на которой установлен кодовый замок.

На пульте замка есть d кнопок с цифрами $0, 1, \ldots, d-1$. Известно, что код, открывающий замок, состоит из k цифр. Замок открывается, если последние k нажатий кнопок образуют код.

Поскольку Петя не имеет понятия, какой код открывает замок, ему придётся перебрать все возможные коды из k цифр. Но, чтобы как можно скорее попасть на тренировку, нужно минимизировать количество нажатий на кнопки. Помогите Пете придумать такую последовательность нажатий на кнопки, при которой все возможные коды были бы проверены, а количество нажатий при этом оказалось бы минимально возможным.

Формат входных данных

В первой строке входного файла записаны через пробел два целых числа d и k — количество кнопок на пульте и размер кода, соответственно ($2 \leqslant d \leqslant 10$, $1 \leqslant k \leqslant 20$).

Формат выходных данных

В первой строке выходного файла выведите искомую последовательность. Если последовательностей минимальной длины, перебирающих все возможные коды, несколько, можно выводить любую из них. Гарантируется, что d и k таковы, что минимальная длина последовательности не превосходит 1 мебибайта.

Пример

codelock.in	codelock.out
2 3	0001011100

Пояснение к примеру

Последовательность в примере перебирает все коды длины 3 в следующем порядке: 000, 001, 010, 101, 011, 111, 110, 100.