

Содержание

Обязательные задачи	2
Задача А. 2-раскраска [1 sec, 256 mb]	2
Задача В. Коммивояжёр возвращается! [4 sec, 256 mb]	3
Задача С. Общий предок [2 sec, 256 mb]	4
Обычные задачи	5
Задача D. Хроматическое число [1 sec, 256 mb]	5
Задача E. Макс клика [1 sec, 512 mb]	6
Задача F. LCA Problem Revisited [4 sec, 256 mb]	7
Дополнительные задачи	8
Задача G. Длинная дорога [4 sec, 256 mb]	8
Задача H. Дерево [5 sec, 512 mb]	9
Задача I. Опекуны карнотавров [2 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**.

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

Задачи расположены в **произвольном порядке**!

Обязательные задачи

Задача А. 2-раскраска [1 сек, 256 mb]

Выясните, можно ли правильно раскрасить данный неориентированный граф в два цвета. Если можно, то приведите правильную раскраску.

Формат входных данных

В первой строке входного файла два целых числа N и M ($1 \leq N \leq 2 \cdot 10^4$, $0 \leq M \leq 2 \cdot 10^4$) — число вершин и рёбер в графе.

В следующих M строках заданы рёбра графа, каждое двумя целыми числами от 1 до N — номерами его концов. Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

Если правильно раскрасить граф в два цвета невозможно, выведите “NO”.

Иначе выведите “YES” и N целых чисел — цвета вершин в порядке от 1-й до N -й. Цвет каждой вершины должен быть 1 или 2.

Примеры

stdin	stdout
4 3 1 2 3 2 1 4	YES 1 2 1 2
3 3 1 2 1 3 3 2	NO

Задача В. Коммивояжёр возвращается! [4 sec, 256 mb]

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

Формат входных данных

В системе Альфы Центавра n планет. Это число записано в первой строке входного файла ($1 \leq n \leq 19$). Следующие n строк содержат по n чисел каждая: j -ое число на i -ой из этих строк — стоимость перемещения a_{ij} от i -ой планеты до j -ой. Числа в каждой строке разделены пробелами. Числа a_{ii} не несут полезной информации. Все числа во входном файле положительны и не превосходят 10^8 .

Формат выходных данных

В первой строке выходного файла выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите n чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

Пример

stdin	stdout
3	5
8 1 6	3 1 2
3 5 7	
4 9 2	

Задача С. Общий предок [2 сек, 256 mb]

Дано подвешенное дерево с корнем в 1-й вершине и M запросов вида “найти у двух вершин наименьшего общего предка”.

Формат входных данных

В первой строке файла записано одно число N — количество вершин. В следующих $N - 1$ строках записаны числа. Число x на строке $2 \leq i \leq N$ означает, что x — отец вершин i . ($x < i$). На следующей строке число M . Следующие M строк содержат запросы вида (x, y) — найти наименьшего предка вершин x и y . Ограничения: $1 \leq N \leq 5 \cdot 10^4, 0 \leq M \leq 5 \cdot 10^4$.

Формат выходных данных

M ответов на запросы.

Пример

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

Обычные задачи

Задача D. Хроматическое число [1 сек, 256 mb]

Хроматическое число графа — такое минимальное число k , что граф можно правильно раскрасить в k цветов. Найдите хроматическое число данного графа.

Формат входных данных

В первой строке входного файла два целых числа N и M ($1 \leq N \leq 13$, $0 \leq M \leq 78$) — число вершин и рёбер в графе.

В следующих M строках заданы рёбра графа, каждое двумя целыми числами от 1 до N — номерами его концов. Гарантируется, что в графе нет петель и кратных рёбер.

Формат выходных данных

Выведите искомое хроматическое число.

Примеры

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

Задача Е. Макс клика [1 sec, 512 mb]

Макс, опытейший игрок в доту, постоянно кликал.

Дан случайный неориентированный граф G из n вершин и m ребер.
Подкликой называется такое подмножествоо вершин A : $\forall a, b \in A, a \neq b \quad \exists$ ребро (a, b) .
Ваша задача — найти подклику A : $|A|$ максимально.

Формат входных данных

На первой строке число вершин $1 \leq n \leq 40$ и число ребер $m \geq 1$.
Следующие m строк содержат пары чисел от 1 до n — ребра графа.
В графе нет ни петель, ни кратных ребер.

Формат выходных данных

На первой строке выведите k — количество вершин в максимальной подклике. На следующей строке k целых чисел от 1 до n — номера вершин в подклике. Вершины можно выводить в любом порядке. Если максимальных подклик несколько, выведите любую.

Примеры

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 1 3 4 5

Задача F. LCA Problem Revisited [4 sec, 256 mb]

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 100\,000$) вершин, пронумерованных от 0 до $n - 1$. Требуется ответить на m ($1 \leq m \leq 10\,000\,000$) запросов о наименьшем общем предке для пары вершин.

Запросы генерируются следующим образом. Заданы числа a_1, a_2 и числа x, y и z . Числа a_3, \dots, a_{2m} генерируются следующим образом: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. Первый запрос имеет вид $\langle a_1, a_2 \rangle$. Если ответ на $i - 1$ -й запрос равен v , то i -й запрос имеет вид $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$.

Формат входных данных

Первая строка содержит два числа: n и m . Корень дерева имеет номер 0. Вторая строка содержит $n - 1$ целых чисел, i -е из этих чисел равно номеру родителя вершины i . Третья строка содержит два целых числа в диапазоне от 0 до $n - 1$: a_1 и a_2 . Четвертая строка содержит три целых числа: x, y и z , эти числа неотрицательны и не превосходят 10^9 .

Формат выходных данных

Выведите в выходной файл сумму номеров вершин — ответов на все запросы.

Примеры

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

Дополнительные задачи

Задача G. Длинная дорога [4 sec, 256 mb]

Дорога, дорога, осталось немного...

Дан случайный неориентированный граф G из n вершин и m ребер. Ваша задача — найти гамильтонов путь. Гарантируется, что гамильтонов путь в графе есть.

Формат входных данных

На первой строке число вершин $n \leq 50$ и число ребер $m \geq 1$.

Следующие m строк содержат пары чисел от 1 до n — ребра графа.

В графе нет ни петель, ни кратных ребер.

Поскольку почти полный граф — совсем не интересный тест, $m \leq 500$.

Формат выходных данных

На первой строке выведите n различных чисел от 1 до n — вершины гамильтонового пути в порядке прохода по ним. Начинать и заканчивать можно в любой вершине. Если гамильтоновых путей несколько, выведите любой.

Примеры

stdin	stdout
5 8 3 1 2 5 5 4 3 4 1 4 3 5 3 2 1 2	1 4 3 5 2

Задача Н. Дерево [5 sec, 512 mb]

Задано подвешенное дерево, содержащее n ($1 \leq n \leq 1\,000\,000$) вершин. Каждая вершина покрашена в один из n цветов. Требуется для каждой вершины v вычислить количество различных цветов, встречающихся в поддереве с корнем v .

Формат входных данных

В первой строке входного файла задано число n . Последующие n строк описывают вершины, по одной в строке. Описание очередной вершины i имеет вид $p_i \ c_i$, где p_i — номер родителя вершины i , а c_i — цвет вершины i ($1 \leq c_i \leq n$). Для корня дерева $p_i = 0$.

Формат выходных данных

Выведите n чисел, обозначающих количества различных цветов в поддеревьях с корнями в вершинах $1, \dots, n$.

Примеры

stdin	stdout
5	1 2 3 1 1
2 1	
3 2	
0 3	
3 3	
2 1	

Задача I. Опекуны карнотавров [2 сек, 256 mb]

Карнотавры очень внимательно относятся к заботе о своем потомстве. У каждого динозавра обязательно есть старший динозавр, который его опекает. В случае, если опекуна съедают (к сожалению, в юрский период такое не было редкостью), забота о его подопечных ложится на плечи того, кто опекал съеденного динозавра. Карнотавры — смертоносные хищники, поэтому их обычаи строго запрещают им драться между собой. Если у них возникает какой-то конфликт, то, чтобы решить его, они обращаются к кому-то из старших, которому доверяют, а доверяют они только тем, кто является их опекуном или опекуном их опекуна и так далее (назовем таких динозавров суперопекунами). Поэтому для того, чтобы решить спор двух карнотавров, нужно найти такого динозавра, который является суперопекуном для них обоих. Разумеется, беспокоить старших по пустякам не стоит, поэтому спорщики стараются найти самого младшего из динозавров, который удовлетворяет этому условию. Если у динозавра возник конфликт с его суперопекуном, то этот суперопекун сам решит проблему. Если у динозавра нелады с самим собой, он должен разобраться с этим самостоятельно, не беспокоя старших. Помогите динозаврам разрешить их споры.

Формат входных данных

Во входном файле записано число M , обозначающее количество запросов ($1 \leq M \leq 200\,000$). Далее на отдельных строках следуют M запросов, обозначающих следующие события:

- $+ v$ — родился новый динозавр и опекунство над ним взял динозавр с номером v . Родившемуся динозавру нужно присвоить наименьший натуральный номер, который до этого еще никогда не встречался.
- $- v$ — динозавра номер v съели.
- $? u v$ — у динозавров с номерами u и v возник конфликт и вам надо найти им третейского судью.

Изначально есть один прадинозавр номер 1; гарантируется, что он никогда не будет съеден.

Формат выходных данных

Для каждого запроса типа «?» в выходной файл нужно вывести на отдельной строке одно число — номер самого молодого динозавра, который может выступить в роли третейского судьи.

Примеры

stdin	stdout
11	1
+ 1	1
+ 1	2
+ 2	2
? 2 3	5
? 1 3	
? 2 4	
+ 4	
+ 4	
- 4	
? 5 6	
? 5 5	