Содержание

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 A. Pairs. Паросочетание [0.4 sec, 256 mb]

Двудольным графом называется граф $(V, E), E \subset V \times V$ такой, что его множество вершин V можно разбить на два подмножества A и B, для которых $\forall (e_1, e_2) \in E\ e_1 \in A, e_2 \in B$ и $A, B \subset E, A \cap B = \varnothing$.

Паросочетанием в двудольном графе называется любой его набор несмежных ребер, то есть такой набор $S \subset E$, что для любых двух ребер $e_1 = (u_1, v_1), e_2 = (u_2, v_2)$ из S выполнено $u_1 \neq u_2$ и $v_1 \neq v_2$.

Ваша задача — найти максимальное паросочтание в двудольном графе, то есть паросочетание с максимально возможным числом ребер.

Формат входных данных

В первой строке записаны два целых числа n и m ($1 \leq n, m \leqslant 250$) — число вершин в A и число вершин в B.

Далее следуют n строк с описаниями ребер. i-я вершина из A описана в $i + 1$-й строке файла. Каждая из этих строк содержит номера вершин из B, соединенных с i-й вершиной A. Вершины в A и B нумеруются независимо (с единицы). Список завершается числом 0.

Формат выходных данных

Первая строка выходного файла должна содержать одно целое число l — количество ребер в максимальном паросочетании. Далее должны следовать l строк, в каждой из которых должны быть два целых числа u_j и v_j — концы ребер паросочетания в A и B, соотвественно.

Пример

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

Задача B. Общий предок [0.5 sec, 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 дереве.

В следующих $n - 1$ строках записаны числа. Число x на строке i означает, что x — непосредственный предок вершины i. $(x < i)$.

Далее следует одно число m — количество запросов. Каждый запрос содержит две вершины x и y.

Ограничения: $2 \leqslant n \leqslant 5 \cdot 10^4, 0 \leqslant m \leqslant 5 \cdot 10^4$.

Формат выходных данных

Для каждого запроса в отдельной строке выведите наименьшего общего предка вершин из этого запроса.

Пример

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

Обычные задачи

Задача C. Минимальное контролирующее множество [1 sec, 256 mb]

Требуется построить в двудольном графе минимальное контролирующее множество, если дано максимальное паросочетание.

Формат входных данных

В первой строке файла даны два числа m и n ($1 \leqslant m, n \leqslant 4\,000$) — размеры долей. Каждая из следующих m строк содержит список ребер, выходящих из соответствующей вершины первой доли. Этот список начинается с числа K_i ($0 \leqslant K_i \leqslant n$) — количества ребер, после которого записаны вершины второй доли, соединенные с данной вершиной первой доли, в произвольном порядке. Сумма всех K_i во входном файле не превосходит $500\,000$. Последняя строка файла содержит некоторое максимальное паросочетание в этом графе — m чисел $0 \leqslant L_i \leqslant n$ — соответствующая i-й вершине первой доли вершина второй доли, или 0, если i-я вершина первой доли не входит в паросочетание.

Формат выходных данных

Первая строка содержит размер минимального контролирующего множества. Вторая строка содержит количество вершин первой доли S, после которого записаны S чисел — номера вершин первой доли, входящих в контролирующее множество, в возрастающем порядке. Третья строка содержит описание вершин второй доли в аналогичном формате.

Пример

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

Задача D. Такси [0.4 sec, 256 mb]

Управлять службой такси — сосвем не простое дело. Помимо естественной необходимости централизованного управления машинами для того, чтобы обслуживать заказы по мере их поступления и как можно быстрее, нужно также планировать поездки для обслуживания тех клиентов, которые сделали заказы заранее.

В вашем распоряжении находится список заказов такси на следующий день. Вам необходимо минимимизировать число машин такси, необходимых чтобы выполнить все заказы.

Для простоты будем считать, что план города представляет собой квадратную решетку. Адрес в городе будем обозначать парой целых чисел: x-координатой и y-координатой. Время, необходимое для того, чтобы добраться из точки с адресом (a, b) в точку (c, d), равно $|a-c|+|b-d|$ минут. Машина такси может выполнить очередной заказ, либо если это первый ее заказ за день, либо она успевает приехать в начальную точку из предыдущей конечной хотя бы за минуту до указанного срока. Обратите внимание, что выполнение некоторых заказов может окончиться после полуночи.

Формат входных данных

В первой строке входного файла записано число заказов M ($0 < M < 500$). Последующие M строк описывают сами заказы, по одному в строке. Про каждый заказ указано время отправления в формате `hh:mm` (в интервале с `00:00` по `23:59`), координаты (a, b) точки отправления и координаты (c, d) точки назначения. Все координаты во входном файле неотрицательные и не превосходят 200. Заказы записаны упорядоченными по времени отправления.

Формат выходных данных

В выходной файл выведите единственное целое число — минимальное количество машин такси, необходимых для обслуживания всех заказов.

Пример

stdin	stdout
2 08:00 10 11 9 16 08:07 9 16 10 11	1
2 08:00 10 11 9 16 08:06 9 16 10 11	2

Задача E. LCA Problem Revisited [2.2 sec, 256 mb]

Задано подвешенное дерево, содержащее n ($1 \leqslant n \leqslant 100\,000$) вершин, пронумерованных от 0 до $n-1$. Требуется ответить на m ($1 \leqslant m \leqslant 10\,000\,000$) 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 $a_3, \ldots, a_{2m}$ 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 F. Опекуны карнотавров [2 sec, 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 динамозавра. 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 ($1 \le M \le 200\,000$). 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 + 2 ? 2 3 ? 1 3 ? 2 4 + 4 + 4 - 4 ? 5 6 ? 5 5	1 1 2 2 5

Задача G. Женитьба [3 sec, 256 mb]

Давным давно в одной далёкой стране правил мудрый царь. И было у него ни много, ни мало M дочерей. Вот настало время выдавать дочерей замуж, и послал царь гонцов в N соседних государств. На эту весть съехалось по одному принцу от каждого государства. Так как царь был любящим отцом, учитывающим мнение своих дочерей, первым делом он потребовал принцев выстроиться в ряд, занумеровал юношей числами от 1 до N, и спросил у каждой дочери, с какими из стоящих молодых людей она согласна сыграть свадьбу. У царя этой страны было хорошее математическое образование, и ему не составило бы труда по этой информации проверить, можно ли назначить каждой дочери своего жениха из числа симпатичных ей молодых людей. Но пытливый ум правителя страны заинтересовал такой вопрос: сколько существует пар (L, R) $(1 \leqslant L \leqslant R \leqslant N)$, таких, что из юношей с номерами от L до R включительно можно найти по жениху для каждой из дочерей? Помогите царю найти ответ на его вопрос!

Формат входных данных

В первой строке входного файла заданы три целых числа N, M и K $(1 \leqslant N \leqslant 30\,000$, $1 \leqslant M \leqslant 2\,000$, $1 \leqslant K \leqslant \min(N \times M, 100\,000)$ – соответственно количество юношей, количество девушек и количество строк, описывающих предпочтения девушек. В каждой из следующих K строк записаны два целых чисел A_i, B_i $(1 \leqslant A_i \leqslant N, 1 \leqslant B_i \leqslant M)$, которые означают, что девушке B_i нравится юноша A_i. Все записи различны.

Формат выходных данных

Примеры

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

Замечание

В тесте из условия подходят пары (1, 3), (1, 4), (1, 5) и (2, 5).