

А. Дерево?

Неорієнтований граф без петель та кратних ребер задано матрицею суміжності. Визначити, чи є цей граф деревом.

Вхідні дані

Перший рядок містить кількість вершин графа n ($1 \leq n \leq 100$). Далі записана матриця суміжності розміром $n \times n$, у якій 1 позначає наявність ребра, 0 - його відсутність. Матриця симетрична відносно головної діагоналі.

Вихідні дані

Виведіть повідомлення **YES**, якщо граф є деревом, і **NO** у протилежному випадку.

Приклад введення	Приклад виведення
3 0 1 0 1 0 1 0 1 0	YES

В. Топологічне сортування

Задано орієнтований незважений граф. Необхідно топологічно відсортувати його вершини.

Вхідні дані

У першому рядку міститься кількість вершин n ($1 \leq n \leq 100000$) та кількість ребер m ($1 \leq m \leq 100000$) у графі. У наступних m рядках перелічені ребра графу, кожне з яких задається парою чисел — номерами початкової та кінцевої вершини.

Вихідні дані

Надрукувати довільне топологічне сортування графу у вигляді послідовності номерів вершин.

Якщо граф неможливо топологічно відсортувати вивести "-1".

Приклад введення	Приклад виведення
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

C. Мости

Задано неорієтовний граф. Потрібно знайти усі мости у ньому.

Вхідні дані

Перший рядок містить два натуральних числа n та m ($n \leq 20000$, $m \leq 200000$) - кількість вершин та ребер графа відповідно.

Наступні m рядків містять опис ребер по одному у рядку. Ребро номер i описується двома натуральними числами b_i, e_i ($1 \leq b_i, e_i \leq n$) - номерами вершин, які воно сполучає.

Вихідні дані

Перший рядок повинен містити одне натуральне число b - кількість мостів у заданому графі. У наступному рядку виведіть b цілих чисел - номери ребер, які є мостами, у зростаючому порядку. Ребра нумеруються з одиниці у тому порядку, у якому вони поступають на вхід.

Приклад введення	Приклад виведення
6 7 1 2 2 3 3 4 1 3 4 5 4 6 5 6	1 3

D. Точки сполучення

Задано неорієнтовний граф. Знайти усі точки сполучення у ньому.

Вхідні дані

Перший рядок містить два натуральних числа n та m ($n \leq 20000$, $m \leq 200000$) - кількість вершин та ребер графа відповідно.

Наступні m рядків містять описи ребер по одному у рядку. Ребро номер i описується двома натуральними числами b_i, e_i ($1 \leq b_i, e_i \leq n$) - номерами кінців ребра.

Вихідні дані

Перший рядок повинен містити кількість точок сполучення b у заданому графі. У наступних b рядках виведіть по одному цілому числу - номери вершин, які є точками сполучення, у зростаючому порядку.

Приклад введення	Приклад виведення
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

Е. Максимальне пароутворення

Граф (V, E) називається двудольним, якщо множину його вершин V можна розбити на дві підмножини A і B такі, що будь-яке ребро із E з'єднує вершину із A з вершиною із B .

Пароутворення P називається будь-яка підмножина E така, що ніякі два ребра із нього не має спільної вершини.

Максимальне пароутворення - це пароутворення, кількість ребер в якому максимальна.

Знайдіть максимальне пароутворення в даному двудольному графі.

Вхідні дані

В першому рядку задано три числа n, m і k ($1 \leq n, m \leq 100, 1 \leq k \leq 10000$), де n - кількість вершин в множині A , m - кількість вершин в B , а k - кількість ребер в графі. Кожен з наступних k рядків містить по два числа u_i і v_i , означає, що вершина u_i множини A з'єднана ребром з v_i множини B . Вершини у множинах A і B нумеруються окремо, починаючи з одиниці. Всі вхідні числа цілі.

Вихідні дані

В першому рядку виведіть кількість I ребер в максимальному пароутворення. Далі виведіть I рядків, по два числа в кожному. Числа a_j і b_j , що стоять в j -ой із цих рядків, означає, що пароутворення взято між вершиною a_j множини A і вершиною b_j множини B . Взяті ребра повинні утворити максимальне пароутворення.

Приклад введення	Приклад виведення
2 3 4 1 1 1 2 2 2 2 3	2 1 1 2 3

Е. Погодні умови

Система рейсів авіакомпанії **OlympAirways** була спроектована таким чином, щоб з будь-якого аеропорту, що обслуговується авіакомпанією, можна було перелетіти до будь-якого іншого аеропорту, скориставшись, можливо, більше ніж одним рейсом. Кожен рейс сполучає два аеропорти, та виконується у обидва боки.

Існує проблема, що деякі рейси певний час можуть не виконуватись через погані погодні умови. Таким чином, ймовірно, що клієнт не зможе перелетіти з аеропорту **A** до **B**, користуючись лише літаками авіакомпанії **OlympAirways**. Для дослідження подібних ситуацій науковий відділ компанії ввів поняття числа вразливості зв'язку між парою аеропортів **A** та **B**. Це число дорівнює кількості рейсів авіакомпанії, відміна довільного з яких (при умові, що всі інші рейси виконуються у звичайному порядку) призведе до неможливості перельоту до аеропорту **B** з аеропорту **A**.

Напишіть програму, яка за інформацією про усі рейси, що виконуються авіакомпанією, визначає суму чисел вразливості зв'язку між усіма парами аеропортів.

Вхідні дані

Перший рядок містить ціле число **N** ($1 \leq N \leq 100$) - кількість аеропортів, що обслуговуються авіакомпанією. Другий рядок містить ціле число **M** ($1 \leq M \leq 4950$) - кількість рейсів, які виконуються авіакомпанією. Кожний з наступних **M** рядків визначає рейс, який представлено парою цілих чисел від **1** до **N** - номерами аеропортів, які він сполучає.

Вихідні дані

Вивести одне ціле число - сумарне число вразливості зв'язку між усіма різними парами аеропортів **A** та **B**, таких, що номер **A** менше за номер **B**.

Приклад введення	Приклад виведення
5 5 1 2 4 2 4 5 3 2 3 1	10

G. Пошук циклу

Дано орієнтований незважений граф. Необхідно визначити, чиє в ньому цикли. І якщо є, то вивести будь-який з них.

Вхідні дані

У першому рядку знаходяться два натуральні числа n та m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$) - кількість вершин та ребер у графі відповідно. Далі в m рядках перелічені ребра графу. Кожне ребро задається парою чисел - номерами початкової та кінцевої вершини відповідно.

Вихідні дані

Якщо в графі немає циклу, то вивести **"NO"**, інакше вивести **"YES"** і далі перелічити вершини у порядку обходу циклу.

Приклад введення	Приклад виведення
2 2 1 2 2 1	YES 1 2
2 2 1 2 1 2	NO

Н. АВТОГОНКИ

У місті **N** у найближчий час відбудеться етап чемпіонату світу з автогонок серед автомобілів класу Формула-0. Оскільки спеціальний автодром для цих змагань організатори побудувати не встигли, було вирішено організувати трасу на вулицях міста.

У місті **N** є **n** перехресть, деякі пари з яких з'єднані дорогами, рух по яких можливий в обох напрямках. При цьому довільні два перехрестя з'єднані не більше ніж однією дорогою, і є можливість доїхати по дорогах від довільного перехрестя до довільного іншого.

Траса, на якій будуть проводитись змагання, повинна бути круговою (тобто повинна починатись і завершуватись на одному й тому ж перехресті), при цьому в процесі руху по ній ніяке перехрестя не повинно зустрічатись більше одного разу.

На попередньому етапі підготовки оргкомітетом було створено список всіх доріг міста. Тепер настав час його використати. Перше питання, яке необхідно вирішити, - це питання про існування у місті потрібної кругової траси (зрозуміло, якщо відповідь буде негативною, організаторам доведеться у терміновому порядку побудувати ще декілька доріг). Єдина проблема полягає у тому, що у організаторів є підозра, що, оскільки список складвся не дуже уважно, у ньому деякі дороги вказані більше одного разу.

Напишіть програму, яка за заданим списком доріг міста визначить, чи можлива організація в місті потрібної кругової траси.

Вхідні дані

Перший рядок містить два цілих числа: **n** ($1 \leq n \leq 1000$) - кількість перехресть у місті **N** і **m** ($0 \leq m \leq 100000$) - кількість доріг у складеному списку.

Наступні **m** рядків описують дороги. Кожна дорога описується двома числами: **u** і **v** ($1 \leq u, v \leq n, u \neq v$) - номери перехресть, які вона з'єднує. Так як дороги двосторонні, то пара чисел (**u**, **v**) і пара чисел (**v**, **u**) описують одну й ту ж дорогу.

Вихідні дані

Вивести **YES**, якщо у місті можна організувати кругову трасу для змагань, і слово **NO** - у протилежному випадку.

Приклад введення	Приклад виведення
3 4 1 2 2 3 3 1 3 2	YES
2 3 1 2 2 1 2 1	NO

I. Зараз

Зараз літо, літня школа закінчується. І лише спогади про минуле заставляють тверезо дивитись на речі.

— *Невже це кінець?*

— *Хто знає...*

— *Ну тоді скоріше до справи!*

Задано неорієнтовний граф без петель та кратних ребер. Знайти величину максимального пароутворення, тобто максимальний розмір підмножини P ребер графа, що довільній вершині інцидентно не більше одного ребра з P .

Вхідні дані

У першому рядку задано два числа N ($1 \leq N \leq 400$) та K ($0 \leq K \leq N \cdot (N-1)/2$) — кількість вершин і ребер у графі. Кожен з наступних K рядків містить по два числа u і v — опис одного ребра. Гарантується, що граф цілком випадковий.

Вихідні дані

Вивести єдине число — величину максимального пароутворення.

Приклад введення	Приклад виведення
5 5 1 2 1 3 1 4 1 5 2 3	2

J. Відстань на дереві

Деревом називається зв'язаний граф, який не містить циклів.

Відстанню між двома вершинами дерева називається довжина (ребрах) найкоротшого шляху між цими вершинами.

Дано дерево з n вершин і додатне число k . Знайдіть кількість різних пар вершин дерева, відстань між якими дорівнює k . Зверніть увагу, що пари (v, u) і (u, v) вважаються однією і тією ж парою.

Вхідні дані

В першому рядку записано два цілих числа n і k ($1 \leq n \leq 50000$, $1 \leq k \leq 500$) - кількість вершин дерева і задана відстань, між вершинами.

В наступних $n - 1$ рядках записано ребра дерева у форматі $a_i \ b_i$ ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), де a_i і b_i - вершини дерева, з'єднані i -им ребром. Всі задані ребра різні.

Вихідні дані

Виведіть одне ціле число - кількість різних пар вершин дерева, відстань між якими дорівнює k .

Приклад введення	Приклад виведення
5 2 1 2 2 3 3 4 2 5	4

К. Кольорові чарівники

Казкова країна являє собою множину міст, з'єднаних дорогами з двостороннім рухом. Причому з довільного міста країни можна дістатись у довільне інше місто або безпосередньо, або через інші міста. Відомо, що у казковій країні не існує доріг, які з'єднують місто саме з собою і між довільними двома різними містами, існує не більше однієї дороги.

У казковій країні живуть жовтий та синій чарівники. Жовтий чарівник, пройшовши по дорозі, перефарбовує її у жовтий колір, синій — у синій. Як відомо, при накладанні жовтої фарби на синю, або синьої фарбки на жовту, фарби змішуються і перетворюються у фарбку зеленого кольору, який є самим нелюбимим кольором обох чарівників.

У цьому році у столиці країни (місті f) проводиться конференція чарівників. Тому жовтий та синій чарівники хочуть взнати, яку мінімальну кількість доріг їм доведеться перефарбувати у зелений колір, щоб дістатись до столиці. На початку усі дороги не пофарбовані.

Початкове положення жовтого та синього чарівників наперед не відомо. Тому необхідно розв'язати дану задачу для k можливих випадків їхніх початкових розміщень.

Вхідні дані

Перший рядок містить кількість міст n ($1 \leq n \leq 100$) та доріг m ($1 \leq m \leq 500$) у чарівній країні відповідно. Третій рядок містить номер міста f ($1 \leq f \leq n$), яке є столицею казкової країни. У наступних m рядках знаходиться опис доріг країни. У цих m рядках записано по два цілих числа a_i та b_i , які означають, що існує дорога, яка з'єднує міста a_i та b_i . Наступний рядок містить кількість k ($1 \leq k \leq 100$) можливих початкових розміщень чарівників. Далі йде k рядків, кожен з яких містить два цілих числа — номери міст, у яких спочатку знаходяться жовтий та синій чарівники відповідно.

Вихідні дані

Для кожного з k випадків слід вивести мінімальну кількість доріг, які доведеться пофарбувати у зелений колір чарівникам для того, щоб дістатись до столиці.

Приклад введення	Приклад виведення
6 6 1 1 2 2 3 3 4 4 2 4 5 3 6 2 5 6 6 6	1 2

L. Комерційний проект

Компанія «UltraSoft» має N працівників. Кожен працівник має лише одного керівника підрозділу. Декілька підрозділів мають одного керівника, над якими є свої керівники і так до генерального директора.

Декілька працівників різних підрозділів під час форуму компанії придумали як оптимізувати проект, що розробляє компанія. Але щоб узгодити дану розробку потрібно, щоб кожен працівник узгодив проект зі своїм керівником підрозділу, а кожен керівник підрозділу зі своїм керівником. Проект буде узгоджений, якщо ланцюг узгоджень дійде до спільного керівника всіх працівників які придумали оптимізацію.

Скільки працівників та керівників потрібно задіяти, щоб узгодити новацію?

Вхідні дані:

В першому рядку записано число N — загальна кількість працівників компанії «UltraSoft» ($1 \leq N \leq 10^5$). Далі описана ієрархія працівників компанії.

В другому рядку записано $N-1$ число, номери керівників кожного працівника від 2-го до n -го.

Номер генерального директора компанії — 1.

В третьому рядку записано число K — кількість працівників, які придумали новацію. В четвертому рядку записані номери цих працівників.

Вихідні дані:

Вивести одне число — кількість працівників компанії, які повинні бути задіяні в узгодженні проекту.

Приклад введення	Приклад виведення
7 1 1 3 3 5 5 2 4 6	4

М. Предок

Напишіть програму, яка для двох вершин дерева визначить, чи є одна з них предком іншої.

Вхідні дані

Перший рядок містить кількість вершин у дереві n ($1 \leq n \leq 100000$). У другому рядку знаходиться n чисел, i -те з яких визначає номер безпосереднього батька вершини з номером i . Якщо це число дорівнює нулю, то вершина є коренем дерева.

У третьому рядку знаходиться кількість запитів m ($1 \leq m \leq 100000$). Кожен з наступних m рядків містить два різних числа a та b ($1 \leq a, b \leq n$).

Вихідні дані

Для кожного з m запитів виведіть у окремому рядку число **1**, якщо вершина a є одним з предків вершини b , і **0** у протилежному випадку.

Приклад введення	Приклад виведення
6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5	0 1 1 0 0

Н. Кольорові чарівники - 2

Казкова країна являє собою множину міст, з'єднаних дорогами з двостороннім рухом. Причому з довільного міста країни можна дістатись в довільне інше місто або безпосередньо, або через інші міста. Відомо, що в казковій країні не існує доріг, що з'єднують місто саме з собою, та між довільними двома різними містами існує не більше однієї дороги.

В казковій країні живуть жовтий та синій чарівники. Жовтий чарівник, проходячи по дорозі, перефарбовує її в жовтий колір, синій — в синій. Як відомо, при накладанні жовтої фарби на синюю, або синьої фарби на жовту, фарби змішуються і перетворюються в фарбу зеленого кольору, яка є самим нелюбимим кольором обох чарівників.

В цьому році в столиці країни (місті **F**) проводиться конференція чарівників. Тому жовтий і синій чарівники хочуть взнати, яку мінімальну кількість доріг їм прийдеться перефарбувати в зелений колір, щоб дістатись в столицю. Спочатку всі дороги не пофарбовані.

Початкове положення жовтого та синього чарівників наперед не відомо. Тому потрібно розв'язати дану задачу для k можливих випадків їх початкових розміщень.

Вхідні дані

Перший рядок містить цілі числа: n ($1 \leq n \leq 10^5$) та m ($1 \leq m \leq 500000$) - кількість міст та доріг в чарівній країні відповідно. Третій рядок містить одне ціле число F ($1 \leq F \leq n$) - номер міста, що є столицею казкової країни. В наступних M рядках міститься опис доріг країни. В цих m рядках записано по два цілих числа a_i та b_i , які означають, що існує дорога, що з'єднує міста a_i та b_i . Наступний рядок містить ціле число k ($1 \leq k \leq 10^5$) - кількість можливих початкових розміщень чарівників. Далі йде K рядків, кожен з яких містить два цілих числа - номери міст, в яких на початку знаходяться жовтий та синій чарівники відповідно.

Вихідні дані

Для кожного з k тестів вивести мінімальну кількість доріг, які прийдеться пофарбувати в зелений колір чарівникам для того щоб дістатися до столиці.

Приклад введення	Приклад виведення
6 6 1 1 2 2 3 3 4 4 2 4 5 3 6 2 5 6 6 6	1 2