

Problem A. Khóa học online 1

Time Limit 1000 ms

Mem Limit 524288 kB

You have to complete n courses. There are m requirements of the form "course a has to be completed before course b ". Your task is to find an order in which you can complete the courses.

Input

The first input line has two integers n and m : the number of courses and requirements.

The courses are numbered $1, 2, \dots, n$.

After this, there are m lines describing the requirements. Each line has two integers a and b : course a has to be completed before course b .

Output

Print an order in which you can complete the courses. You can print any valid order that includes all the courses.

If there are no solutions, print "IMPOSSIBLE".

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

Input	Output
5 3 1 2 3 1 4 5	3 4 1 5 2

Problem B. Quick sort

Time Limit	500 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Sandro là một người rất có tổ chức. Mỗi ngày, anh ấy lập danh sách những việc cần phải làm và đánh số chúng từ 1 đến n . Tuy nhiên, một số việc cần phải được thực hiện trước những việc khác. Trong bài toán này, bạn phải tìm ra liệu Sandro có thể hoàn thành tất cả các nhiệm vụ của mình hay không và nếu có, hãy in ra thứ tự đúng.

Input

Trong dòng đầu tiên, bạn được cung cấp một số nguyên n và m ($1 \leq n \leq 10000$, $1 \leq m \leq 1000000$). Trong m dòng tiếp theo có hai số nguyên khác nhau x và y , ($1 \leq x, y \leq 10000$) mô tả rằng công việc x cần phải được thực hiện trước công việc y .

Output

In "Sandro fails." nếu Sandro không thể hoàn thành tất cả các nhiệm vụ trong danh sách. Nếu có một giải pháp, hãy in ra thứ tự đúng, các công việc cần được thực hiện cách nhau bởi một khoảng trắng. Nếu có nhiều giải pháp, hãy in ra giải pháp mà số đầu tiên nhỏ nhất, nếu vẫn còn nhiều giải pháp, hãy in ra giải pháp mà số thứ hai nhỏ nhất, và cứ thế tiếp tục.

Example 1

Input	Output
8 9 1 4 1 2 4 2 4 3 3 2 5 2 3 5 8 2 8 6	1 4 3 5 7 8 2 6

Example 2

Input	Output
2 2 1 2 2 1	Sandro fails.

Problem C. scc

Time Limit 380 ms
Mem Limit 524288 kB

给定有向图 $G(V, E)$ ，其中有 N ($1 \leq N \leq 10^4$) 个顶点和 M ($1 \leq M \leq 10^5$) 条边，请计算 G 的强连通分量数。

输入

第一行包含两个整数 N, M 。
接下来的 M 行描述了 G 的一条边。

输出

输出一个整数，表示强连通分量的数量。

Input	Output
3 2 1 2 2 3	3
Input	Output
3 3 1 2 2 3 3 1	1

注意

你可以参考这里的算法: [Tarjan Algorithm](#)

John has n tasks to do. Unfortunately, the tasks are not independent and the execution of one task is only possible if other tasks have already been executed.

Input

The input will consist of several instances of the problem. Each instance begins with a line containing two integers, $1 \leq n \leq 100$ and m . n is the number of tasks (numbered from 1 to n) and m is the number of direct precedence relations between tasks. After this, there will be m lines with two integers i and j , representing the fact that task i must be executed before task j .

An instance with $n = m = 0$ will finish the input.

Output

For each instance, print a line with n integers representing the tasks in a possible order of execution.

Sample Input

```
5 4
1 2
2 3
1 3
1 5
0 0
```

Sample Output

```
1 4 2 5 3
```

Order is an important concept in mathematics and in computer science. For example, Zorn's Lemma states: "a partially ordered set in which every chain has an upper bound contains a maximal element." Order is also important in reasoning about the fix-point semantics of programs.

This problem involves neither Zorn's Lemma nor fix-point semantics, but does involve order.

Given a list of variable constraints of the form $x < y$, you are to write a program that prints all orderings of the variables that are consistent with the constraints.

For example, given the constraints $x < y$ and $x < z$ there are two orderings of the variables x , y , and z that are consistent with these constraints: xyz and xzy .

Input

The input consists of a sequence of constraint specifications. A specification consists of two lines: a list of variables on one line followed by a list of constraints on the next line. A constraint is given by a pair of variables, where ' $x\ y$ ' indicates that $x < y$.

All variables are single character, lower-case letters. There will be at least two variables, and no more than 20 variables in a specification. There will be at least one constraint, and no more than 50 constraints in a specification. There will be at least one, and no more than 300 orderings consistent with the constraints in a specification.

Input is terminated by end-of-file.

Output

For each constraint specification, all orderings consistent with the constraints should be printed. Orderings are printed in lexicographical (alphabetical) order, one per line.

Output for different constraint specifications is separated by a blank line.

Sample Input

```
a b f g
a b b f
v w x y z
v y x v z v w v
```

Sample Output

```
abfg
abgf
agbf
gabf
```

```
wxzvy
wzxvy
xwzvy
xzwvy
zwxvy
zxwvy
```

A rare book collector recently discovered a book written in an unfamiliar language that used the same characters as the English language. The book contained a short index, but the ordering of the items in the index was different from what one would expect if the characters were ordered the same way as in the English alphabet. The collector tried to use the index to determine the ordering of characters (i.e., the collating sequence) of the strange alphabet, then gave up with frustration at the tedium of the task.

You are to write a program to complete the collector's work. In particular, your program will take a set of strings that has been sorted according to a particular collating sequence and determine what that sequence is.

Input

The input consists of an ordered list of strings of uppercase letters, one string per line. Each string contains at most 20 characters. The end of the list is signalled by a line with the single character '#'. Not all letters are necessarily used, but the list will imply a complete ordering among those letters that are used.

Output

Your output should be a single line containing uppercase letters in the order that specifies the collating sequence used to produce the input data file.

Sample Input

```
XWY
ZX
ZXY
ZXW
YWWX
#
```

Sample Output

```
XZYW
```

Problem G. Rank sort

Time Limit	1000 ms
Mem Limit	1572864 kB
Code Length Limit	50000 B
OS	Linux

Eloy is a hard worker man, however, he is constantly bullied by his superiors, molested by this, one day he was wondering in what “rank” you are, so you can bully the people with lower ranks, also to discover who can really bully Eloy!.

Now, given the number of employees and the number of relations between them, Eloy need you to output the “rank” which employee is in, being 1 the “boss” (not bullied by anybody) and the employee who are in these ranks

Input

There will be an integer T denoting the number of test cases, then, T test cases will follow. Each test case starts with two integers N and R, the number of employees and the number of relations between them. The next R lines consists of two integers R1 and R2, meaning that “employee R1 is lower than employee R2's rank”.

Output

You will output for each test case the string “Scenario #i:” where i is the test case you are analyzing, after that, you will print N lines, for each line you will output the rank of the employee and the employee itself, if there is the same rank for several employees, then output them lexicographically ordered (the first is the lower.)

Sample

Input	Output
2 5 6 2 0 2 4 1 4 1 2 3 2 4 0 5 4 1 0 2 0 3 2 4 2	Scenario #1: 1 0 2 4 3 2 4 1 4 3 Scenario #2: 1 0 2 1 2 2 3 3 3 4

Blank line between test cases for clarification and separation. Please note that can be more than one "boss" (not bullied by anybody.)

Constraints

$1 \leq N \leq 1000$; $1 \leq R \leq 10000$

Problem H. DP Lex 1

Time Limit 2000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Fox Ciel is going to publish a paper on FOCS (Foxes Operated Computer Systems, pronounce: "Fox"). She heard a rumor: the authors list on the paper is always sorted in the lexicographical order.

After checking some examples, she found out that sometimes it wasn't true. On some papers authors' names weren't sorted in lexicographical order in normal sense. But it was always true that after some modification of the order of letters in alphabet, the order of authors becomes lexicographical!

She wants to know, if there exists an order of letters in Latin alphabet such that the names on the paper she is submitting are following in the lexicographical order. If so, you should find out any such order.

Lexicographical order is defined in following way. When we compare s and t , first we find the leftmost position with differing characters: $s_i \neq t_i$. If there is no such position (i. e. s is a prefix of t or vice versa) the shortest string is less. Otherwise, we compare characters s_i and t_i according to their order in alphabet.

Input

The first line contains an integer n ($1 \leq n \leq 100$): number of names.

Each of the following n lines contain one string $name_i$ ($1 \leq |name_i| \leq 100$), the i -th name.

Each name contains only lowercase Latin letters. All names are different.

Output

If there exists such order of letters that the given names are sorted lexicographically, output any such order as a permutation of characters 'a' – 'z' (i. e. first output the first letter of the modified alphabet, then the second, and so on).

Otherwise output a single word "Impossible" (without quotes).

Examples

Input	Output
3 rivest shamir adleman	bcdefghijklmnopqrsatuvwxyz

Input	Output
10 tourist petr wjmzbr yeputons vepifanov scottwu oooooooooooooooooooo subscriber rowdark tankengineer	Impossible

Input	Output
10 petr egor endagorion feferivan ilovetanyaromanova kostka dmitriyh maratsnowbear bredorjaguarturnik cgyforever	aghjlnopefikdmbcqrstuvwxyz

Input	Output
7 car care careful carefully becarefuldontforgetsomething otherwiseyouwillbehacked goodluck	acbdefhijklmnogpqrstuvwxyz

Problem 1. Đường đi dài nhất

Time Limit 2000 ms

Mem Limit 1024 kB

Cho đồ thị có hướng G có N đỉnh và M cạnh. Các đỉnh được đánh số từ 1 tới N , và cạnh có hướng thứ i ($1 \leq i \leq M$) đi từ đỉnh x_i tới đỉnh y_i .

Đảm bảo rằng đồ thị G không có chu trình.

Tìm độ dài đường đi có hướng dài nhất trong đồ thị G , quy ước rằng: **độ dài của một đường đi có hướng** là số cạnh nằm trên đường đi đó.

Input

- Dòng đầu tiên chứa hai số nguyên N và M , lần lượt là số đỉnh và số cạnh của đồ thị đã cho. ($2 \leq N \leq 10^5, 1 \leq M \leq 10^5$)
- Dòng thứ i trong số M dòng tiếp theo, mỗi dòng chứa hai số nguyên x_i và y_i , biểu diễn cho cạnh thứ i của đồ thị.

Đảm bảo rằng tất cả các cạnh (x_i, y_i) bất kì đều phân biệt và đồ thị G không có chu trình.

Outout

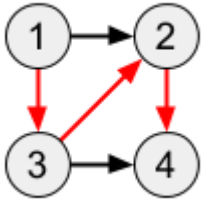
In ra độ dài của đường đi có hướng dài nhất trong đồ thị G .

Sample 1

Input	Output
<pre> 4 5 1 2 1 3 3 2 2 4 3 4 </pre>	3

Giải thích

Đường màu đỏ trong hình dưới đây là đường đi dài nhất:

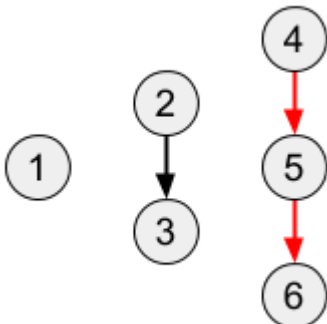


Sample 2

Input	Output
6 3 2 3 4 5 5 6	2

Giải thích

Đường màu đỏ trong hình dưới đây là đường đi dài nhất:

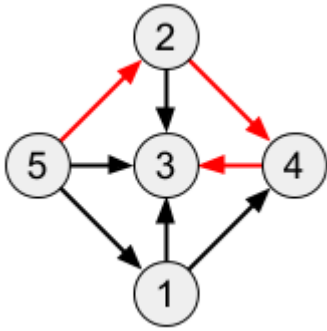


Sample 3

Input	Output
5 8 5 3 2 3 2 4 5 2 5 1 1 4 4 3 1 3	3

Giải thích

Đường màu đỏ trong hình dưới đây là đường đi dài nhất:



Problem J. Dijkstra

Time Limit 1000 ms

Mem Limit 524288 kB

Uolevi has won a contest, and the prize is a free flight trip that can consist of one or more flights through cities. Of course, Uolevi wants to choose a trip that has as many cities as possible.

Uolevi wants to fly from Syrjälä to Lehmälä so that he visits the maximum number of cities. You are given the list of possible flights, and you know that there are no directed cycles in the flight network.

Input

The first input line has two integers n and m : the number of cities and flights. The cities are numbered $1, 2, \dots, n$. City 1 is Syrjälä, and city n is Lehmälä.

After this, there are m lines describing the flights. Each line has two integers a and b : there is a flight from city a to city b . Each flight is a one-way flight.

Output

First print the maximum number of cities on the route. After this, print the cities in the order they will be visited. You can print any valid solution.

If there are no solutions, print "IMPOSSIBLE".

Constraints

- $2 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

Input	Output
5 5 1 2 2 5 1 3 3 4 4 5	4 1 3 4 5

DATE

Trong thành phố Freecontest có m con đường một chiều nối ngôi nhà a với ngôi nhà b . Naul là một tên tra nam có n cô bạn gái. Biết cô bạn gái thứ i sẽ ở ngôi nhà thứ i . Mặc dù vậy, Naul chỉ thích hẹn hò với cô thứ 1 và cô thứ n nên Naul cần biết có bao nhiêu cách để đi từ nhà cô thứ nhất đến nhà cô thứ n . Hãy trở thành người bạn tốt của Naul và giúp Naul tìm xem có bao nhiêu cách nhé!

Dữ liệu

- Dòng thứ nhất ghi hai số nguyên n, m ($1 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$) – lần lượt là số cô bạn gái và số lượng con đường một chiều.
- m dòng tiếp theo, dòng thứ i gồm hai số nguyên a và b ($1 \leq a, b \leq n$) – con đường nối ngôi nhà a với ngôi nhà b . Dữ liệu luôn đảm bảo không tồn tại cặp chỉ số (i, j) nào mà cả hai cô gái i và j đều có thể đi đến nhà của nhau

Kết quả

In ra số cách để đi từ nhà cô thứ nhất đến nhà cô thứ n . Kết quả lấy dư theo module $10^9 + 7$

Ví dụ

Sample Input	Sample Output
4 5 1 2 2 4 1 3 3 4 1 4	3

Problem L. Counting sort

Input File milkorder.in

Output File milkorder.out

Bò của nông dân John ($1 \leq N \leq 10^5$), được đánh số từ $1 \dots N$ như mọi khi, tình cờ có quá nhiều thời gian trên chân của họ. Kết quả là, họ đã xây dựng một cấu trúc xã hội phức tạp liên quan đến thứ tự mà nông dân John vắt sữa chúng mỗi buổi sáng.

Sau vài tuần nghiên cứu, nông dân John đã đưa ra M quan sát về cấu trúc xã hội của bò của mình ($1 \leq M \leq 50,000$). Mỗi quan sát là một danh sách được sắp xếp của một số bò của ông, cho biết rằng những con bò này nên được vắt sữa theo thứ tự mà chúng xuất hiện trong danh sách này. Ví dụ, nếu một trong những quan sát của nông dân John là danh sách 2, 5, 1, nông dân John nên vắt sữa bò 2 vào một thời điểm nào đó trước khi vắt sữa bò 5, người nên được vắt sữa vào một thời điểm nào đó trước khi vắt sữa bò 1.

Các quan sát của nông dân John được ưu tiên, vì vậy mục tiêu của ông là tối đa hóa giá trị của X để thứ tự vắt sữa của ông đáp ứng các điều kiện được đề cập trong X quan sát đầu tiên. Nếu có nhiều thứ tự vắt sữa thỏa mãn các điều kiện đầu tiên này, nông dân John tin rằng đó là một truyền thống lâu đời rằng những con bò có số thấp hơn sẽ xếp trên những con có số cao hơn, vì vậy ông muốn vắt sữa những con bò có số thấp nhất trước. Cụ thể hơn, nếu có nhiều thứ tự vắt sữa thỏa mãn các điều kiện này, nông dân John muốn sử dụng thứ tự từ điển nhỏ nhất. Một thứ tự x nhỏ hơn từ điển so với một thứ tự y nếu với một số j , $x_i = y_i$ cho tất cả $i < j$ và $x_j < y_j$ (nói cách khác, hai thứ tự này giống nhau cho đến một điểm nhất định, tại đó x nhỏ hơn y).

Xin hãy giúp nông dân John xác định thứ tự tốt nhất để vắt sữa bò của ông.

ĐỊNH DẠNG NHẬP (file milkorder.in):

Dòng đầu tiên chứa N và M . M dòng tiếp theo mỗi dòng mô tả một quan sát. Dòng $i + 1$ mô tả quan sát i , và bắt đầu bằng số lượng con bò m_i được liệt kê trong quan sát tiếp theo là danh sách m_i số nguyên cho thứ tự của các con bò trong quan sát. Tổng của m_i là tối đa 200,000.

ĐỊNH DẠNG ĐẦU RA (file milkorder.out):

Đầu ra N số nguyên cách nhau bằng khoảng trắng, cho một hoán vị của $1 \dots N$ chứa thứ tự mà nông dân John nên vắt sữa bò của ông.

VÍ DỤ NHẬP:

4 3
3 1 2 3
2 4 2
3 3 4 1

VÍ DỤ ĐẦU RA:

1 4 2 3

Ở đây, nông dân John có bốn con bò và nên vắt sữa bò 1 trước bò 2 và bò 2 trước bò 3 (quan sát đầu tiên), bò 4 trước bò 2 (quan sát thứ hai) và bò 3 trước bò 4 và bò 4 trước bò 1 (quan sát thứ ba). Hai quan sát đầu tiên có thể được thỏa mãn đồng thời, nhưng nông dân John không thể đáp ứng tất cả các tiêu chí này cùng một lúc, vì điều đó sẽ yêu cầu bò 1 đứng trước bò 3 và bò 3 đứng trước bò 1.

Điều này có nghĩa là có hai thứ tự có thể: 1 4 2 3 và 4 1 2 3, thứ tự đầu nhỏ hơn theo thứ tự từ điển.

Tín dụng vấn đề: Jay Leeds

Problem M. DFS

Time Limit 1000 ms

Mem Limit 524288 kB

A game has n levels, connected by m teleporters, and your task is to get from level 1 to level n . The game has been designed so that there are no directed cycles in the underlying graph. In how many ways can you complete the game?

Input

The first input line has two integers n and m : the number of levels and teleporters. The levels are numbered $1, 2, \dots, n$.

After this, there are m lines describing the teleporters. Each line has two integers a and b : there is a teleporter from level a to level b .

Output

Print one integer: the number of ways you can complete the game. Since the result may be large, print it modulo $10^9 + 7$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

Input	Output
4 5 1 2 2 4 1 3 3 4 1 4	3

Problem N. Hash sort

Time Limit 2000 ms

Mem Limit 262144 kB

Bạn được cho n mẫu p_1, p_2, \dots, p_n và m chuỗi s_1, s_2, \dots, s_m . Mỗi mẫu p_i bao gồm k ký tự là chữ cái thường Latin hoặc ký tự đại diện cho bất kỳ ký tự nào (được ký hiệu bằng dấu gạch dưới). Tất cả các mẫu đều khác nhau. Mỗi chuỗi s_j bao gồm k chữ cái thường Latin.

Một chuỗi a khớp với mẫu b nếu với mỗi i từ 1 đến k , hoặc là b_i là ký tự đại diện hoặc $b_i = a_i$.

Bạn được yêu cầu sắp xếp lại các mẫu sao cho mẫu đầu tiên mà chuỗi thứ j khớp là $p[mt_j]$. Bạn được phép để thứ tự của các mẫu không thay đổi.

Bạn có thể thực hiện sắp xếp lại như vậy không? Nếu có, hãy in ra bất kỳ thứ tự hợp lệ nào.

Nhập

Dòng đầu tiên chứa ba số nguyên n, m và k ($1 \leq n, m \leq 10^5, 1 \leq k \leq 4$) — số lượng mẫu, số lượng chuỗi và độ dài của mỗi mẫu và chuỗi.

Mỗi trong n dòng tiếp theo chứa một mẫu — k ký tự là chữ cái thường Latin hoặc dấu gạch dưới. Tất cả các mẫu đều khác nhau.

Mỗi trong m dòng tiếp theo chứa một chuỗi — k chữ cái thường Latin, và một số nguyên mt ($1 \leq mt \leq n$) — chỉ số của mẫu đầu tiên mà chuỗi tương ứng nên khớp.

Đầu ra

In ra "NO" nếu không có cách nào để sắp xếp lại các mẫu sao cho mẫu đầu tiên mà chuỗi thứ j khớp là $p[mt_j]$.

Nếu có, in ra "YES" ở dòng đầu tiên. Dòng thứ hai nên chứa n số nguyên phân biệt từ 1 đến n — thứ tự của các mẫu. Nếu có nhiều câu trả lời, in ra bất kỳ câu trả lời nào.

Ví dụ 1

Input	Output
5 3 4 _b_d __b_ aaaa ab__ _bcd abcd 4 abba 2 dbcd 5	YES 3 2 4 5 1

Ví dụ 2

Input	Output
1 1 3 __c cba 1	NO

Ví dụ 3

Input	Output
2 2 2 a_ __b_ ab 1 ab 2	NO

Ghi chú

Thứ tự của các mẫu sau khi sắp xếp lại trong ví dụ đầu tiên là như sau:

- aaaa
- __b_
- ab__
- _bcd
- _b_d

Do đó, chuỗi đầu tiên khớp với các mẫu ab__, _bcd, _b_d theo thứ tự, mẫu đầu tiên là ab__, đúng là $p[4]$. Chuỗi thứ hai khớp với __b_ và ab__, mẫu đầu tiên là __b_, đúng là $p[2]$. Chuỗi cuối khớp với _bcd và _b_d, mẫu đầu tiên là _bcd, đúng là $p[5]$.

Câu trả lời cho bài kiểm tra này không duy nhất, còn các thứ tự hợp lệ khác cũng tồn tại.

Trong ví dụ thứ hai, cba không khớp với __c, do đó, không tồn tại thứ tự hợp lệ.

Trong ví dụ thứ ba, thứ tự $(a_ , _b)$ khiến cả hai chuỗi khớp với mẫu 1 đầu tiên và thứ tự $(_b, a_)$ khiến cả hai chuỗi khớp với mẫu 2 đầu tiên. Do đó, không có thứ tự nào tạo ra kết quả 1 và 2.

Problem O. Khóa học online 2

Time Limit 1000 ms

Mem Limit 524288 kB

Bạn đang muốn tham gia vào n khóa học online. Có tất cả m yêu cầu được đặt ra có dạng “khóa học a phải được hoàn thành trước khóa học b ”. Ngoài ra, bạn phải hoàn thành khóa học thứ 1 sớm nhất có thể. Nếu có cách để thực hiện thì bạn cũng phải hoàn thành khóa học thứ 2 sớm nhất có thể. Nói cách khác, hãy tìm dãy thứ tự hoàn thành các khóa học có thứ tự từ điển nhỏ nhất

Yêu cầu: Hãy tìm ra thứ tự các khóa học để bạn có thể hoàn thành tất cả các khóa học online mà bạn muốn tham gia.

INPUT

- Dòng đầu tiên chứa hai số nguyên n, m .
- m dòng tiếp theo mỗi dòng chứa hai số nguyên a, b thể hiện một yêu cầu.

OUTPUT

- In ra trên một dòng là thứ tự các khóa học mà bạn cần tìm.
- Dữ liệu đảm bảo sẽ có ít nhất một cách sắp xếp thứ tự các khóa học.

Ràng buộc

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

Input	Output
4 2 2 1 2 3	2 1 3 4

Problem P. NP hãd

Time Limit 3000 ms

Mem Limit 1048576 kB

Problem Statement

You are given a directed graph with N vertices and M edges. For $i = 1, 2, \dots, M$, the i -th edge is directed from vertex s_i to vertex t_i .

Determine whether there is a permutation $P = (P_1, P_2, \dots, P_N)$ of $(1, 2, \dots, N)$ that satisfies both of the following conditions, and if there is, provide an example.

- $P_{s_i} < P_{t_i}$ for all $i = 1, 2, \dots, M$.
- $L_i \leq P_i \leq R_i$ for all $i = 1, 2, \dots, N$.

Constraints

- $2 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq \min\{4 \times 10^5, N(N-1)\}$
- $1 \leq s_i, t_i \leq N$
- $s_i \neq t_i$
- $i \neq j \implies (s_i, t_i) \neq (s_j, t_j)$
- $1 \leq L_i \leq R_i \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

```
N M
s1 t1
s2 t2
⋮
sM tM
L1 R1
L2 R2
```


\vdots
 $L_N \ R_N$

Output

If there is no P that satisfies the conditions, print **No**. If there is a P that satisfies the conditions, print **Yes** in the first line, and the elements of P separated by spaces in the second line, in the following format. If multiple P 's satisfy the conditions, any of them will be accepted.

Yes
 $P_1 \ P_2 \ \dots \ P_N$

Sample 1

Input	Output
5 4 1 5 2 1 2 5 4 3 1 5 1 3 3 4 1 3 4 5	Yes 3 1 4 2 5

$P = (3, 1, 4, 2, 5)$ satisfies the conditions. In fact,

- for the first edge $(s_1, t_1) = (1, 5)$, we have $P_1 = 3 < 5 = P_5$;
- for the second edge $(s_2, t_2) = (2, 1)$, we have $P_2 = 1 < 3 = P_1$;
- for the third edge $(s_3, t_3) = (2, 5)$, we have $P_2 = 1 < 5 = P_5$;
- for the fourth edge $(s_4, t_4) = (4, 3)$, we have $P_4 = 2 < 4 = P_3$.

Also,

- $L_1 = 1 \leq P_1 = 3 \leq R_1 = 5$,
- $L_2 = 1 \leq P_2 = 1 \leq R_2 = 3$,
- $L_3 = 3 \leq P_3 = 4 \leq R_3 = 4$,
- $L_4 = 1 \leq P_4 = 2 \leq R_4 = 3$,
- $L_5 = 4 \leq P_5 = 5 \leq R_5 = 5$.

Sample 2

Input	Output
2 2 1 2 2 1 1 2 1 2	No

No P satisfies the conditions, so print **No**.

Problem Q. TST 2025

Time Limit 1000 ms

Mem Limit 524288 kB

A game has n rooms and m tunnels between them. Each room has a certain number of coins. What is the maximum number of coins you can collect while moving through the tunnels when you can freely choose your starting and ending room?

Input

The first input line has two integers n and m : the number of rooms and tunnels. The rooms are numbered $1, 2, \dots, n$.

Then, there are n integers k_1, k_2, \dots, k_n : the number of coins in each room.

Finally, there are m lines describing the tunnels. Each line has two integers a and b : there is a tunnel from room a to room b . Each tunnel is a one-way tunnel.

Output

Print one integer: the maximum number of coins you can collect.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \cdot 10^5$
- $1 \leq k_i \leq 10^9$
- $1 \leq a, b \leq n$

Example

Input	Output
4 4 4 5 2 7 1 2 2 1 1 3 2 4	16

Problem R. DP Lex 2

Time Limit 1000 ms

Mem Limit 65536 kB

Mirko có một mảng chứa N từ **khác nhau**, và Mirko muốn mã hoá những từ trong mảng đó sử dụng một phương pháp mã hoá.

Chúng ta mã hoá đoạn văn bản bằng cách chọn một mã khoá – một bảng hoán vị của các chữ cái trong bảng chữ cái tiếng Anh. Chúng ta thay toàn bộ ký tự **a** trong văn bản bằng chữ cái thứ nhất của mã khoá, thay toàn bộ ký tự **b** bằng chữ cái thứ hai của mã khoá, và cứ tiếp tục như vậy tới ký tự **z**.

Ngoài mảng trên ra, Mirko còn có một mảng A chứa các số nguyên từ 1 tới N theo một thứ tự nhất định nào đó. Mirko muốn chọn một mã khoá sao cho mảng chứa các từ sau khi được mã hoá và sắp xếp theo thứ tự từ điển phải tương ứng với mảng A . Chính xác hơn, Mirko muốn từ có vị trí ban đầu ở A_i phải có vị trí i sau khi được mã hoá và sắp xếp.

Thứ tự từ điển là thứ tự mà các từ xuất hiện trong từ điển. Để so sánh thứ tự của hai từ, chúng ta so sánh lần lượt các ký tự từ trái qua phải của hai từ đó để xác định xem từ nào xuất hiện trước. Nếu như từ **Y** bắt đầu bằng từ **X**, thì ta nói từ **X** có thứ tự nhỏ hơn từ **Y**.

Mirko đang cảm thấy không muốn tự mã hoá mảng văn bản của mình, vậy nên Mirko muốn nhờ bạn làm việc này.

Input

- Dòng đầu tiên chứa số nguyên N ($2 \leq N \leq 100$).
- N dòng tiếp theo mỗi dòng chứa một từ (dài nhất 100 ký tự tiếng Anh viết thường). Các từ không trùng lặp.
- Dòng cuối chứa N số nguyên – các thành phần của mảng A .

Output

- Trong trường hợp không có mã khoá tương thích, in **NE**.
- Trong trường hợp có mã khoá tương thích, in **DA** ở dòng thứ nhất, ở dòng thứ hai in ra đoạn mã khoá được sử dụng (một từ gồm 26 ký tự khác nhau trong bảng chữ cái tiếng Anh).

- Trong trường hợp có nhiều hơn một mã khoá tương thích, in ra một đoạn mã khoá bất kì.

Input	Output
2 ab bc 2 1	DA bacdefghijklmnopqrstuvwxyz
Input	Output
3 abc bcd add 1 2 3	NE
Input	Output
3 bbb ccc ddd 2 3 1	DA adbcefg hijklmnopqrstuvwxyz

Giải thích ví dụ 1

Sau khi mã hoá, các từ trở thành **ba**, **ac**, và sau khi sắp xếp thứ tự, mảng trở thành **ac**, **ba**, nghĩa là từ thứ nhất sau khi sắp xếp có vị trí thứ hai, và từ thứ hai sau khi sắp xếp có vị trí thứ nhất.

Giải thích ví dụ 3

Sau khi mã hoá, các từ trở thành **ddd**, **bbb**, **ccc**, và sau khi sắp xếp thứ tự, mảng trở thành **bbb**, **ccc**, **ddd**, nghĩa là từ thứ nhất sau khi sắp xếp có vị trí thứ ba, từ thứ ba ở vị trí thứ hai, và từ thứ hai sau khi sắp xếp có vị trí thứ nhất.

Problem S. Viettel Post

Time Limit 2000 ms

Mem Limit 262144 kB

Input File stdin

Output File stdout

Your city has n junctions. There are m **one-way** roads between the junctions. As a mayor of the city, you have to ensure the security of all the junctions.

To ensure the security, you have to build some police checkpoints. Checkposts can only be built in a junction. A checkpost at junction i can protect junction j if either $i = j$ or the police patrol car can go to j from i and then come back to i .

Building checkposts costs some money. As some areas of the city are more expensive than others, building checkpost at some junctions might cost more money than other junctions.

You have to determine the minimum possible money needed to ensure the security of all the junctions. Also you have to find the number of ways to ensure the security in minimum price and **in addition in minimum number of checkposts**. Two ways are different if any of the junctions contains a checkpost in one of them and do not contain in the other.

Input

In the first line, you will be given an integer n , number of junctions ($1 \leq n \leq 10^5$). In the next line, n space-separated integers will be given. The i^{th} integer is the cost of building checkpost at the i^{th} junction (costs will be non-negative and will not exceed 10^9).

The next line will contain an integer m ($0 \leq m \leq 3 \cdot 10^5$). And each of the next m lines contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$; $u \neq v$). A pair u_i, v_i means, that there is a one-way road which goes from u_i to v_i . There will not be more than one road between two nodes in the same direction.

Output

Print two integers separated by spaces. The first one is the minimum possible money needed to ensure the security of all the junctions. And the second one is the number of ways you can ensure the security modulo $1000000007 (10^9 + 7)$.

Examples

Input	Output
3 1 2 3 3 1 2 2 3 3 2	3 1

Input	Output
5 2 8 0 6 0 6 1 4 1 3 2 4 3 4 4 5 5 1	8 2

Input	Output
10 1 3 2 2 1 3 1 4 10 10 12 1 2 2 3 3 1 3 4 4 5 5 6 5 7 6 4 7 3 8 9 9 10 10 9	15 6

Input	Output
2 7 91 2 1 2 2 1	7 1

Problem T. An toàn thông tin

Time Limit 380 ms

Mem Limit 524288 kB

Một lớp gồm N học sinh, mỗi học sinh cho biết những bạn mà học sinh đó có thể liên lạc được (chú ý liên lạc này là liên lạc một chiều: u có thể gửi tin tới v nhưng v thì chưa chắc đã có thể gửi tin tới u).

Thầy chủ nhiệm đang có một thông tin rất quan trọng cần thông báo tới tất cả các học sinh. Để tiết kiệm thời gian, thầy chỉ nhắn tin tới 1 số học sinh rồi sau đó nhờ các học sinh này nhắn lại cho tất cả các bạn mà các học sinh đó có thể liên lạc được, và cứ lần lượt như thế làm sao cho tất cả các học sinh trong lớp đều nhận được tin.

Hãy tìm một số ít nhất các học sinh mà thầy chủ nhiệm cần nhắn.

Input

- Dòng đầu là N, M ($N \leq 800$, M là số lượng liên lạc 1 chiều)
- Một số dòng tiếp theo mỗi dòng gồm 2 số u, v cho biết học sinh u có thể gửi tin tới học sinh v

Output

- Gồm 1 dòng ghi số học sinh cần thầy nhắn tin.

Input	Output
12 15 1 3 3 6 6 1 6 8 8 12 12 9 9 6 2 4 4 5 5 2 4 6 7 10 10 11 11 7 10 9	2

Note

Chọn các học sinh 7 và 2.

Problem U. Halloween

Time Limit 380 ms

Mem Limit 524288 kB

Hàng năm ở Wisconsin tụi bò lại tổ chức ngày hội Halloween vào kỳ nghỉ Thu. Chúng sẽ mặc đồ hóa trang và đi xin kẹo nông dân John đã đặt trong N ($1 \leq N \leq 100000$) chuồng bò (để thuận tiện ta đánh số các chuồng bò từ $1 \rightarrow N$).

Để cho lũ bò chơi vui hơn, ở chuồng i John sẽ cắm 1 biển báo $next_i$ ($1 \leq next_i \leq N$) cho biết sau khi xin kẹo ở chuồng i thì bò sẽ phải tiếp tục đi tới chuồng $next_i$ để xin kẹo tiếp.

Bò i sẽ bắt đầu xin kẹo từ chuồng i . Và một con bò sẽ dừng việc xin kẹo nếu nó đi tới 1 chuồng mà nó đã từng đi qua rồi.

Tính xem mỗi con bò sẽ xin được bao nhiêu kẹo, biết rằng ở mỗi chuồng chúng chỉ xin được 1 viên kẹo mà thôi.

Input

- Dòng 1: Một số nguyên duy nhất: N
- Dòng 2 ... $N + 1$: Dòng $i + 1$ gồm 1 số nguyên duy nhất: $next_i$

Output

- Dòng 1 ... N : Dòng i chứa 1 số nguyên là số kẹo mà bò i nhận được

Input	Output
4	1
1	2
3	2
2	3
3	



4

Commuter Pass

JOI-kun is living in a city with N stations. The stations are numbered from 1 to N . There are M railways numbered from 1 to M . The railway i ($1 \leq i \leq M$) connects the station A_i and the station B_i in both directions, and the fare is C_i yen.

JOI-kun is living near the station S , and goes to the IOI high school near the station T . He is planning to buy a commuter pass connecting these two stations. When he buys a commuter pass, he needs to choose a route between the station S and the station T with minimum cost. Using this commuter pass, he can take any railways contained in a chosen route in any directions without additional costs.

JOI-kun often goes to bookstores near the station U and the station V . Therefore, he wants to buy a commuter pass so that the cost from the station U to the station V is minimized.

When he moves from the station U to the station V , he first choose a route from the station U to the station V . Then the fare he has to pay is

- 0 yen if the railway i is contained in a route chosen when he buys a commuter pass, or
- C_i yen if the railway i is not contained in a route chosen when he buys a commuter pass.

The sum of the above fare is the cost from the station U to the station V .

He wants to know the minimum cost from the station U to the station V if he chooses a route appropriately when he buys a commuter pass.

Task

Write a program which calculates the minimum cost from the station U to the station V if he chooses a route appropriately when he buys a commuter pass.

Input

Read the following data from the standard input.

- The first line of input contains two space separated integers N, M . This means the city JOI-kun lives in has N stations and M railways.
- The second line contains two space separated integers S, T . This means JOI-kun is planning to buy a commuter pass from the station S to the station T .
- The third line contains two space separated integers U, V . This means JOI-kun wants to minimize the cost from the station U to the station V .
- The i -th line ($1 \leq i \leq M$) of the following M lines contains three space separated integers A_i, B_i, C_i . The railway i connects the station A_i and the station B_i in both directions, and the fare is C_i yen.



Output

Write one line to the standard output. The output should contain the minimum cost from the station U to the station V if he chooses a route appropriately when he buys a commuter pass.

Constraints

All input data satisfy the following conditions.

- $2 \leq N \leq 100\,000$.
- $1 \leq M \leq 200\,000$.
- $1 \leq S \leq N$.
- $1 \leq T \leq N$.
- $1 \leq U \leq N$.
- $1 \leq V \leq N$.
- $S \neq T$.
- $U \neq V$.
- $S \neq U$ or $T \neq V$.
- JOI-kun can move from any stations to any other stations taking railways.
- $1 \leq A_i < B_i \leq N$ ($1 \leq i \leq M$).
- For every $1 \leq i < j \leq M$, either $A_i \neq A_j$ or $B_i \neq B_j$.
- $1 \leq C_i \leq 1\,000\,000\,000$ ($1 \leq i \leq M$).

Subtask

Subtask 1 [16 points]

- $S = U$.

Subtask 2 [15 points]

- There is a unique route with minimum cost from the station S to the station T .



Subtask 3 [24 points]

- $N \leq 300$.

Subtask 4 [45 points]

- There are no additional constraints.

Sample Input and Output

Sample Input 1	Sample Output 1
6 6 1 6 1 4 1 2 1 2 3 1 3 5 1 2 4 3 4 5 2 5 6 1	2

In this sample input, there is only one route JOI-kun can choose when he buys a commuter pass: Station 1 → Station 2 → Station 3 → Station 5 → Station 6.

In order to minimize the cost from the station 1 to the station 4, he chooses the following route: Station 1 → Station 2 → Station 3 → Station 5 → Station 4. When he chooses this route, the fare he has to pay is

- 2 yen for the railway 5 connecting the station 4 and the station 5, and
- 0 yen for other railways.

Hence the total cost is 2 yen.



Sample Input 2	Sample Output 2
6 5 1 2 3 6 1 2 1000000000 2 3 1000000000 3 4 1000000000 4 5 1000000000 5 6 1000000000	30000000000

In this sample input, JOI-kun does not use the commuter pass when he moves from the station 3 to the station 6.

Sample Input 3	Sample Output 3
8 8 5 7 6 8 1 2 2 2 3 3 3 4 4 1 4 1 1 5 5 2 6 6 3 7 7 4 8 8	15

Sample Input 4	Sample Output 4
5 5 1 5 2 3 1 2 1 2 3 10 2 4 10 3 5 10 4 5 10	0



Sample Input 5	Sample Output 5
10 15 6 8 7 9 2 7 12 8 10 17 1 3 1 3 8 14 5 7 15 2 3 7 1 10 14 3 6 12 1 5 10 8 9 1 2 9 7 1 4 1 1 8 1 2 4 7 5 6 16	19

Problem W. Cười voi

Time Limit 1000 ms

Mem Limit 262144 kB

Trung là một thầy thuốc có tiếng bởi tài năng và lòng nhân từ của mình. Sau kì thi VOI vừa rồi, nhiều bạn trẻ cười VOI lần đầu đã ~~tạch~~ ngã rất đau, vì vậy thầy thuốc Trung cần một lượng lớn nguyên liệu để bào chế ra thuốc để chữa cho các bạn.

Vùng Trung ở có n ngọn núi, được kết nối bởi m con đường một chiều, trên con đường thứ i ban đầu có các cây thuốc cho sản lượng w_i . Sau khi hái thuốc ở trên một con đường, các cây thuốc trên con đường này sẽ dần mọc lại nhưng sản lượng sẽ giảm đi một nửa (làm tròn xuống số nguyên gần nhất). Nhà của Trung ở ngọn núi 1, tìm lộ trình đi hái được nhiều thuốc nhất (có thể kết thúc ở một ngọn núi bất kì).

Input

- Dòng đầu tiên chứa hai số nguyên n, m ($1 \leq n, m \leq 2 \times 10^5$).
- m dòng tiếp theo, mỗi dòng chứa ba số nguyên u, v, w ($1 \leq u, v \leq n; 1 \leq w \leq 10^9$) mô tả một con đường nối giữa hai ngọn núi.

Output

- In ra số lượng thuốc tối đa Trung có thể thu thập được.

Scoring

Subtask	Điểm	Giới hạn
1	30	$n, m \leq 15; w_i = 1 \forall i = 1, 2, \dots, m$
2	30	$u_i \leq v_i$
3	40	Không có ràng buộc gì thêm

Input	Output
5 5 1 3 2 3 4 1 4 2 1 2 1 2 3 5 6	14
Input	Output
6 7 1 4 3 4 5 2 5 1 5 5 2 8 2 3 3 3 6 4 6 2 1	35

Notes

Trong ví dụ, lộ trình Trung có thể đi là $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 5$ với lượng thuốc thu thập được là $2 + 1 + 1 + 2 + 1 + 0 + 0 + 1 + 0 + 6 = 14$.

Problem X. Cam con

Time Limit 380 ms

Mem Limit 524288 kB

Công việc buôn bán dừa của Pirate không mấy khả quan cho lắm, khiến anh đêm ăn không ngon ngày ngủ không yên, chỉ biết chúi đầu vào xem "Rôbô trái cây". Một ngày nọ, đang nằm ngủ dưới gốc dừa, bỗng một trái dừa rơi vào đầu anh ấy. Cũng giống như Newton, Pirate cũng cầm trái dừa lên, ngắm nghía và ...rủa: "Khỉ thật, sao xứ này toàn là dừa thế này!". Tức điên lên, Pirate quyết trồng thêm các loại trái cây khác vào hòn đảo của mình.

Đến mùa thu hoạch, Pirate đặt hàng một "Rôbô trái cây" để giúp mình hái quả. Khu vườn của Pirate có hình chữ nhật, và được chia thành $M \times N$ ô vuông bằng nhau. Trong mỗi ô vuông có một cây thuộc một loại quả khác nhau, đánh số từ 0 đến 9. Không phải vô tình mà chúng được đánh số như vậy, con số đó thể hiện giá trị kinh tế của các loại cây.

Tuy nhiên, nhìn mặt con Rôbô trái cây này có vẻ ngu ngu nên trong lần đầu tiên thử việc, Pirate muốn test AI của nó. Cụ thể là Rôbô phải tuân theo các quy định sau:

1. Tại mỗi ô, Rôbô chỉ có thể đi sang hướng đông hoặc hướng nam sang ô kề cạnh.
2. Có một số ô đặc biệt mà tại đó Rôbô có thể đi được thêm hướng tây hoặc hướng bắc sang ô kề cạnh (chỉ một trong hai).
3. Rôbô không được đi vào những ô có cây dừa (Pirate căm thù dừa).
4. Rôbô được đi qua một ô nhiều lần. Khi đi qua một ô, Rôbô phải hái hết quả ở cây trong ô đó. Lợi nhuận thu được sẽ bằng chỉ số của loại cây vừa được thu hái. Và sau này, không thể đạt thêm lợi nhuận gì từ ô đó nữa.

Xuất phát từ ô ở góc tây bắc của khu vườn, hãy giúp Rôbô trái cây xác định hành trình để đạt được lợi nhuận tối đa.



Input

- Dòng thứ nhất: ghi hai số nguyên M và N - kích thước của khu vườn.
- M dòng tiếp theo: mỗi dòng ghi N kí tự liên tiếp nhau mô tả khu vườn:
 - 'o' - '9': các loại trái cây;
 - '#': cây dừa;
 - 'W': được quyền đi theo hướng tây;
 - 'N': được quyền đi theo hướng bắc.

Output

- Ghi một số nguyên duy nhất là lợi nhuận tối đa đạt được.

Giới hạn

- Trong mọi test, $1 \leq M, N \leq 100$.
- 60% số test có $1 \leq M, N \leq 20$.

Input	Output
2 3 264 3WW	15

Note

Rô-bô sẽ đi theo hành trình như sau $(1, 1) \rightarrow (1, 2) \rightarrow (1, 3) \rightarrow (2, 3) \rightarrow (2, 2) \rightarrow (2, 1)$ (ô (i, j) là ô ở dòng i và cột j).

Tổng lợi nhuận sẽ là $2 + 6 + 4 + 3 = 15$.

Problem Y. Cô hồn thủ đô

Time Limit 4000 ms

Mem Limit 262144 kB

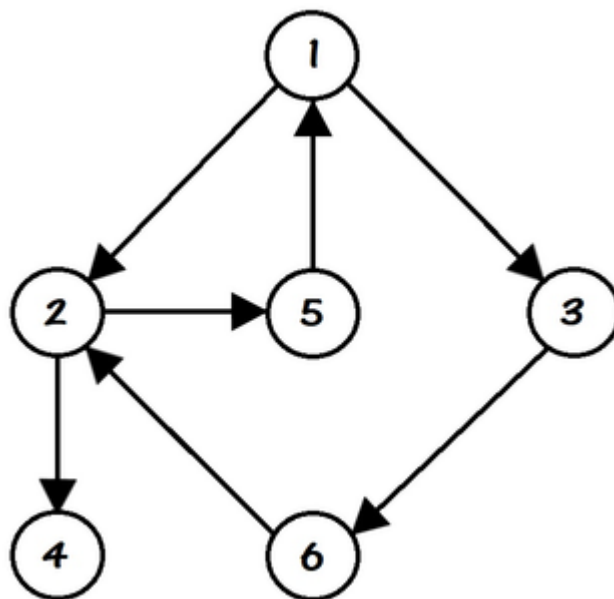
Có n thành phố ở Berland. Thành phố được đánh số 1 là thủ đô. Một số cặp thành phố được kết nối bởi một con đường **một chiều** có độ dài 1.

Trước chuyến đi, Polycarp đã tìm ra giá trị của d_i — khoảng cách ngắn nhất từ thủ đô (thành phố 1) đến thành phố i .

Polycarp bắt đầu hành trình của mình tại thành phố có số s và, khi ở thành phố i , chọn một trong các hành động sau:

1. Di chuyển từ thành phố i đến thành phố j nếu có một con đường từ thành phố i đến thành phố j và $d_i < d_j$;
2. Di chuyển từ thành phố i đến thành phố j nếu có một con đường từ thành phố i đến thành phố j và $d_i \geq d_j$;
3. Dừng lại.

Bởi vì chính phủ của Berland không muốn tất cả mọi người đến thủ đô, nên Polycarp **không quá một lần** có thể thực hiện hành động thứ hai trong danh sách. Nói cách khác, anh ấy có thể thực hiện hành động thứ hai 0 hoặc 1 lần trong suốt hành trình của mình. Polycarp, mặt khác, muốn ở càng gần thủ đô càng tốt.



Ví dụ, nếu $n = 6$ và các thành phố được kết nối, như trong hình trên, thì Polycarp có thể đã thực hiện các chuyến đi sau (không phải tất cả các tùy chọn có thể):

- $2 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 5$;
- $3 \rightarrow 6 \rightarrow 2$;
- $1 \rightarrow 3 \rightarrow 6 \rightarrow 2 \rightarrow 5$.

Polycarp muốn cho mỗi thành phố bắt đầu i biết anh ấy có thể đến gần thủ đô như thế nào. Nói một cách chính thức hơn: anh ấy muốn tìm giá trị tối thiểu của d_j mà Polycarp có thể đạt được từ thành phố i đến thành phố j theo các quy tắc đã mô tả ở trên.

Đầu vào

Dòng đầu tiên chứa một số nguyên t ($1 \leq t \leq 10^4$) — số lượng trường hợp kiểm tra. Sau đó t trường hợp kiểm tra theo sau.

Mỗi trường hợp kiểm tra được trước bởi một dòng trống.

Dòng đầu tiên của mỗi trường hợp kiểm tra chứa hai số nguyên n ($2 \leq n \leq 2 \cdot 10^5$) và m ($1 \leq m \leq 2 \cdot 10^5$) — số lượng thành phố và đường, tương ứng.

Tiếp theo là m dòng mô tả các con đường. Mỗi con đường được đặc trưng bởi hai số nguyên u và v ($1 \leq u, v \leq n, u \neq v$) — số của các thành phố được kết nối bởi một con đường **một chiều**.

Có đảm bảo rằng tổng n và m trên tất cả các trường hợp kiểm tra không vượt quá $2 \cdot 10^5$.

Có đảm bảo rằng đối với mỗi cặp thành phố khác nhau (u, v) có tối đa một con đường từ u đến v (nhưng một cặp đường từ u đến v và từ v đến u — là hợp lệ).

Có đảm bảo rằng có một con đường từ thủ đô đến tất cả các thành phố.

Đầu ra

Đối với mỗi trường hợp kiểm tra, trên một dòng riêng biệt xuất n số, số i -th bằng với khoảng cách tối thiểu có thể từ thủ đô đến thành phố mà Polycarp kết thúc hành trình của mình.

Ví dụ

Input	Output
3	0 0 1 2 0 1
6 7	0 0
1 2	0 0 2 1 1 0
1 3	
2 5	
2 4	
5 1	
3 6	
6 2	
2 2	
1 2	
2 1	
6 8	
1 2	
1 5	
2 6	
6 1	
2 3	
3 4	
4 2	
5 4	