

#Problem 1: MCM Calculation

Description:

You will be given N number of 2D matrices' information aka row and column value for each matrix. You need to calculate the minimum number of operations needed to calculate the multiplication of all the matrices.

In the first line you will be given N. In the following N lines you will find the row and column information of each given matrix, ith line contains the information of ith matrix.

Test Cases:

Input	Output
2 1 2 2 3	6
3 1 2 2 3 3 5	21
3 10 30 30 5 5 40	3500
4 1 5 5 7 7 10 10 5	155
5 1 47 47 5 5 30 30 40 40 23	2505
6 1 7 7 35 35 42 42 3 3 2 2 65	1977

#Problem 2: Unweighted Shortest Path - Undirected

Description: In this problem you will be given an undirected unweighted (or each edge equally weighted) graph G , a source vertex S and a destination vertex D . You need to calculate the shortest distance between S to D . In this problem, the shortest distance between two vertices means using the minimum number of edges to reach from one vertex to the other.

In the first line you will be given two integers V and E denoting the number of vertices and edges of this graph respectively. All the vertices will have ids between 1 to V . In the following E lines, you will get the information about all the edges. In the i^{th} line you will have two integers x ($1 \leq x \leq V$) and y ($1 \leq y \leq V$) which denotes there is an undirected edge between x and y . After E lines of input, you will get two integers S and D ($1 \leq S \leq V$, $1 \leq D \leq V$) denoting the source and destination vertices of the problem.

As output, you will print an integer number d which will denote the minimum distance between S and D . If there lies no path between S and D print "INF" without quotes.

Limits

$1 \leq V \leq 100000$, $1 \leq |E| \leq 100000$

Test Cases:

Input	Output
4 5 1 2 1 3 2 4 3 4 1 4 1 4	1
6 8 1 2 1 3 1 4 1 5 2 5 3 5 4 5 5 6 1 6	2
10 13 1 7 1 5 2 6 2 10	2

2 7 3 8 3 4 3 6 4 6 5 10 6 8 7 9 8 9 2 3	
12 16 1 11 1 8 2 4 2 3 3 7 4 12 4 5 5 10 5 9 6 7 6 10 7 9 7 8 8 12 9 11 11 12 8 10	3
large_in1.txt	

#Problem 3: Bicoloring

Description: In this problem you will be given an undirected unweighted graph G. You have to identify if the given graph is bicolorable or not. The rule of coloring this graph is, for each edge connecting two vertices u and v, must have different colors.

In the first line you will be given two integers V and E denoting the number of vertices and edges of this graph respectively. All the vertices will have ids between 1 to V. In the following E lines, you will get the information about all the edges. In the ith line you will have two integers x ($1 \leq x \leq V$) and y ($1 \leq y \leq V$) which denotes there is an undirected edge between x and y.

As output, you need to print a string "YES" without quotes if the graph is bi-colorable else print "NO".

Limits

$1 \leq V \leq 100000$, $1 \leq |E| \leq 100000$

Test Cases:

Input	Output
4 5 1 2 1 3 2 4 3 4 1 4	NO
4 4 1 2 1 3 2 4 3 4	YES
10 10 1 2 1 9 2 5 3 9 3 4 4 7 5 10 6 7 6 8 8 10	YES
9 9 1 9 1 7 2 6 2 4 3 5 3 7 4 9 5 8 6 8	NO

#Problem 4: Tree Diameter for an undirected tree graph

Description: In this problem you will be given an undirected unweighted Tree graph G. You have to discover the diameter of the given tree. Tree diameter mainly indicates the largest distance found between any two nodes in the given tree. Distance means the number of edges between the nodes.

In the first line you will be given two integers V and E denoting the number of vertices and edges of this graph respectively. All the vertices will have ids between 1 to V. In the following E lines, you will get the information about all the edges. In the i^{th} line you will have two integers x ($1 \leq x \leq V$) and y ($1 \leq y \leq V$) which denotes there is an undirected edge between x and y.

You will write a program that will calculate the diameter of the given undirected tree. For each test case you will print a single line denoting the diameter.

Limits

$1 \leq V \leq 1000$, $1 \leq |E| \leq 1000000$

Test Cases:

Input	Output
5 4 1 2 1 5 2 3 3 4	4
7 6 1 2 1 4 2 3 4 5 5 6 6 7	6
20 19 1 4 2 4 2 5 3 4 4 8 4 6 4 9 4 10 5 17 7 8 11 20 12 17 13 19 14 17 15 17 16 17 17 20 17 19	7

#Problem 5: Multiple shortest path existence

Description: In this problem you will be given an undirected unweighted (or each edge equally weighted) graph G and a source vertex S . You need to find if there exists any node X that can be reached in multiple shortest paths from S . In this problem, the shortest path between two vertices means using the minimum number of edges to reach from one vertex to the other.

In the first line you will be given two integers V and E denoting the number of vertices and edges of this graph respectively. All the vertices will have ids between 1 to V . In the following E lines, you will get the information about all the edges. In the i^{th} line you will have two integers x ($1 \leq x \leq V$) and y ($1 \leq y \leq V$) which denotes there is an undirected edge between x and y . After E lines of input, you will get two a single integer S denoting the source vertex of the problem.

As output, you will either print "YES" (without quotes) if there exists at least one node having multiple shortest paths to reach from S or print "NO" (without quotes) if there exists no such node.

Limits

$1 \leq V \leq 100000$, $1 \leq |E| \leq 100000$

Test Cases:

Input	Output
5 5 1 2 2 3 3 4 1 5 5 3 1	YES
5 5 1 2 2 3 3 4 1 5 4 5 3	NO