



Bookmarks

▶ [How To?](#)▶ [Week 1](#)▶ [Week 2](#)▶ [Week 3](#)▼ [Week 4](#)

[Algorithms on
Graphs](#)

[4th Week
Problems](#)

[due Dec 4, 2016 22:00
CET](#)

[4th Week
Problems: Training](#)

▶ [Week 5](#)

Week 4 > 4th Week Problems > Find a Cycle

Find a Cycle

[Bookmark this page](#)

Find a Cycle

2.0/2.0 points (graded)

Input file:	cycle.in
Output file:	cycle.out
Time limit:	2 seconds
Memory limit:	256 megabytes

The input file contains a description of an unweighted directed graph. You are asked to determine whether this graph contains a cycle. If it does, print any of them.

Input

The first line of the input file contains two integers N and M ($1 \leq N \leq 100\,000, M \leq 100\,000$), the number of vertices and edges in the graph, correspondingly. The following M lines contain descriptions of edges of the graph. Each edge is described by a pair of integers – the indices of the source and target vertex, respectively.

All indices are one-based. The graph may contain loops and multiple edges between the same ordered pair of vertices.

Output

If the graph does not contain a cycle, print “NO”.

Otherwise, print “YES”. In the next line print the vertices which constitute a cycle, in the order along this cycle.

Example

cycle.in	cycle.out
2 2	YES
1 2	2 1
2 1	

2 2	NO
1 2	
1 2	

[Choose Files](#) No file chosen

Accepted

[Submit](#)

You have used 3 of 200 attempts

Discussion

Topic: 08: 4th Week Problems / Find a Cycle

[Show Discussion](#)

© All Rights Reserved



© 2016 edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX®

