

# 1 Checking Consistency of CS Curriculum

## Problem Introduction

A Computer Science curriculum specifies the prerequisites for each course as a list of courses that should be taken before taking this course. You would like to perform a consistency check of the curriculum, that is, to check that there are no cyclic dependencies. For this, you construct the following directed graph: vertices correspond to courses, there is a directed edge  $(u, v)$  if the course  $u$  should be taken before the course  $v$ . Then, it is enough to check whether the resulting graph contains a cycle.

## Problem Description

**Task.** Check whether a given directed graph with  $n$  vertices and  $m$  edges contains a cycle.

**Input Format.** A graph is given in the standard format.

**Constraints.**  $1 \leq n \leq 10^3$ ,  $0 \leq m \leq 10^3$ .

**Output Format.** Output 1 if the graph contains a cycle and 0 otherwise.

**Time Limits.**

language	C	C++	Java	Python	C#	Haskell	JavaScript	Ruby	Scala
time (sec)	1	1	1.5	5	1.5	2	5	5	3

**Memory Limit.** 512MB.

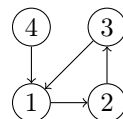
**Sample 1.**

Input:

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

Output:

```
1
```



This graph contains a cycle:  $3 \rightarrow 1 \rightarrow 2 \rightarrow 3$ .

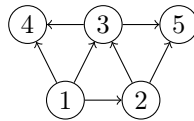
**Sample 2.**

Input:

```
5 7
1 2
2 3
1 3
3 4
1 4
2 5
3 5
```

Output:

```
0
```



There is no cycle in this graph. This can be seen, for example, by noting that all edges in this graph go from a vertex with a smaller number to a vertex with a larger number.