

CSCI 406: AlgoBOWL

Reminder: You are NOT allowed to consult the internet to solve this problem.

1 Problem Description

After an unfortunate accident during E-Days involving Blaster the Burro, a voltmeter, and privilege escalation in notepad: you have to update your class schedule to the most recent catalog. After examining your new classes you notice that you cannot graduate because there are circular dependencies in your class prerequisites. Luckily, you are smart enough to test out of a few classes so you can graduate on time.

Your input is a curriculum of classes labeled 1 to n with prerequisites. The objective is to design an algorithm that finds the minimum number of classes you need to test out of in order to be able to complete your coursework and graduate. As an added note: All of the prerequisites of a course must be completed before you can take that course. Testing out of a required course satisfies the prerequisite requirement of that course for all classes that has that course as a prerequisite.

Input Format:

The input will be provided to you in the following format:

- Lines 1 contains an integer n , the number of courses required to graduate. Lines 2 to $n+1$ list the prerequisites of all the n courses from course $i = 1$ to course $i = n$. Line $i + 1$ specifies an integer m_i , the number of prerequisites, followed by: m_i integers, the prerequisites for course i .

Sample Input:

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

The input above describes a schedule with five classes and the dependencies are shown in Figure 1. The classes are labeled 1 through 5. Class 1 has two prerequisites (class 3 and class 5) and all the other classes have one prerequisite. Observe that in order to take class 1, you need to have taken class 3, to take class 3 you need to have taken class 2, and to take class 2 you have to have taken class 1. This circular dependency can be resolved by testing out of classes 3 and 5; however, a more optimal solution would be to test out of class 1 as this will also remove all circular dependencies.

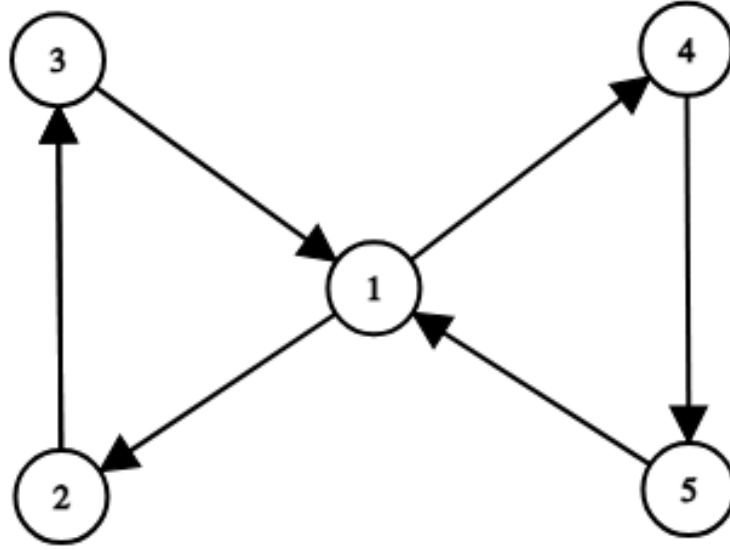


Figure 1: Sample Schedule. This is a graph depicting the dependencies of the curriculum described by the sample input. A directed edge from a vertex a to vertex b indicates that a is a prerequisite of b .

Input Restrictions:

1. $2 \leq n \leq 10^4$
2. $0 \leq m_i \leq n - 1, \forall i, 1 \leq i \leq n$
3. $\sum m_i \leq 10^5$

Output Format: Line 1 of your output will contain the number of classes you have opted to test out of. Line 2 will contain the class labels that you are testing out of. To illustrate, the output for the first and second options presented are as follows:

First option:

2
3 5

Second option:

1
1

Note: This problem is NP-hard, which means that it is unrealistic to expect that your algorithm will compute an optimal solution in a reasonable time frame. Please keep this in mind as you work on this project.

2 Deliverables

Your group has three tasks:

- Develop as good an algorithm as you can that accepts a valid input and produces a valid output.
- Create an input within the parameters specified above that will challenge the other groups.
- Develop a tool that verifies the other groups' outputs. What does this mean? The purpose of the verifier is to examine the outputs that other groups produce on your input and check that the output is valid, i.e., the removal of the specified nodes in the output turns the graph into a DAG. (The verifier is not checking whether the solution provided by the other group is optimal.)

3 Grading Rubric

Points will be distributed according to the following distribution:

- **Valid Input Submitted:** 5 points
- **Input Difficulty:** 10 points
- **Valid Outputs:** 50 points
- **Ranking:** 15 points
- **Verification:** 20 points