Assignment 10

CS329E - Elements of Software Design
Topological Sort
(100 points)
Due Date on Canvas and Gradescope

1 Description

In this assignment, you will implement the Topological Sort and run it on the following graph shown in Figure 1.

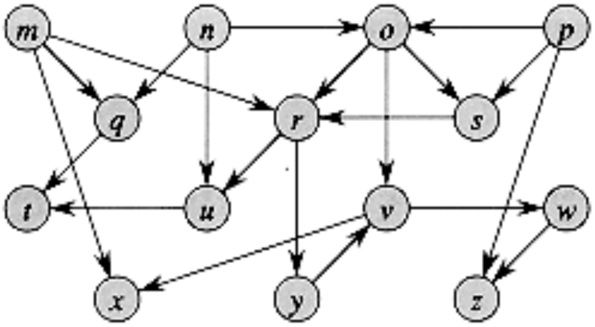

Figure 1: Example directed graph as input data

You will complete the **topo.txt** file that will be used as input for your program. The first couple of edges are given.

- The first line in **topo.txt** will be a single integer *v*. This number will denote the number of vertices to follow.
- The next *v* lines will be the labels for the vertices in alphabetical order.
- There will be one label to a line. The labels are unique.
- The next line after the labels for vertices will be a single number *e*. This number will denote the number of edges to follow. There will be one edge per line.
- Each edge will be of the form - fromVertex and toVertex. Assign a default weight of 1 to each edge.

Here is the outline of the code that we developed in class that you will be modifying. You will use **topo.txt** instead of graph.txt as your input file. You can add an Edge class if you want to. You may use any of the functions that you wrote for graph traversal in your last assignment. You can add more functions as needed. You will first test if the given Graph does not contain a cycle and then do a topological sort on that Graph. For your output, the vertices on a given level must be printed in alphabetical order.

Input:

```
14
m
n
o
p
q
r
s
t
u
v
w
x
y
z
21
m q
m r
m x
n o
n q
n u
...
```

Output:

For the data file given, your output will look as follows:

```
The Graph does not have a cycle.

List of vertices after toposort
['m', 'n', 'p', 'o', 'q', 's', 'r', 'u', 'y', 't', 'v', 'w', 'x', 'z']
```

Pair Programming

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming[1] and abide by the ground rules as stated in that paper. If you are working with a partner then only one of you will be submitting the code. But make sure that your partner's name and UT EID is in the header. If you are working alone then remove the partner's name and eid from the header.

1.1 Turnin

Turn in your assignment on time on Gradescope system on Canvas. For the due date of the assignments, please see the Gradescope and Canvas systems.

1.2 Academic Misconduct Regarding Programming

In a programming class like our class, there is sometimes a very fine line between ”cheating” and acceptable and beneficial interaction between students (In different assignment groups). Thus, it is very important that

[1] Read this paper about Pair Programming `https://collaboration.csc.ncsu.edu/laurie/Papers/Kindergarten.PDF`

you fully understand what is and what is not allowed in terms of collaboration with your classmates. We want to be 100% precise, so that there can be no confusion.

The rule on collaboration and communication with your classmates is very simple: you cannot transmit or receive code from or to anyone in the class in any way – visually (by showing someone your code), electronically (by emailing, posting, or otherwise sending someone your code), verbally (by reading code to someone) or in any other way we have not yet imagined. Any other collaboration is acceptable.

The rule on collaboration and communication with people who are not your classmates (or your TAs or instructor) is also very simple: it is not allowed in any way, period. This disallows (for example) posting any questions of any nature to programming forums such as **StackOverflow**. As far as going to the web and using Google, we will apply the **"two line rule"**. Go to any web page you like and do any search that you like. But you cannot take more than two lines of code from an external resource and actually include it in your assignment in any form. Note that changing variable names or otherwise transforming or obfuscating code you found on the web does not render the "two line rule" inapplicable. It is still a violation to obtain more than two lines of code from an external resource and turn it in, whatever you do to those two lines after you first obtain them.

Furthermore, you should cite your sources. Add a comment to your code that includes the URL(s) that you consulted when constructing your solution. This turns out to be very helpful when you're looking at something you wrote a while ago and you need to remind yourself what you were thinking.

We will use the following Code plagiarism Detection Software to automatically detect plagiarism.

- Staford MOSS

  `https://theory.stanford.edu/~aiken/moss/`

- Jplag - Detecting Software Plagiarism

  `https://github.com/jplag/jplag` and `https://jplag.ipd.kit.edu/`