DSA5101 **Introduction to Big Data for Industry**

Assignment 1

Submission deadline: 31 August 2022

Instructions: 1) The objective of this programming exercise is to train rigorously students in programming in Python. As such, you can only use the basic functions and data types taught in Lectures to write programs. No extra library is allowed in your Python programs if you are not asked to do so.

2). Assignment will AUTOMATICALLY be marked by running a script. Therefore, your programs have to be named as specified in each question. For example, you are asked to name your program for Question 1 as "Q1.py". You must NOT name the program even as "Q1_program.py", "P1.py" or others. This is because our marking pipeline looks for only Q1.py for testing your program. A few marks will be deducted for not using specified names for your programs.

Similarly, the name of input/output files are also specified for each program. You have to use the specified input and output file names when you code and test your programs.

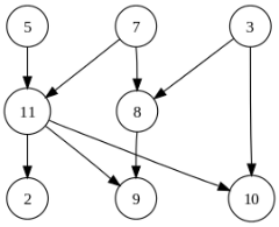
3) We will use our test files to test your programs. These test files will not be given to you before your assignment is submitted. You must test your programs using your own files to make sure your program work properly.

4) To submit your program solution, you include all program files in a directory, compress them into a file named "your-surname.your-firstname.zip" and upload the compressed file onto the directory "Assignment 1" on the module webpage on LumiNUS. You must not have other names.

Q1 (40 Marks) Write a program that outputs the shortest common supersequence (SCS) for two strings that are read from an input file named as "Two_strings.txt" in which each string is given in a line, where each string contains at most 500 English letters. Note that there may be multiple SCSs for two strings. Your program can just output one SCS.

Name your program "Q1.py".

Q2 (30 Marks) A topological sort of a directed graph is a linear ordering of its vertices such that for every directed edge (u, v) from vertex u to vertex v, u comes before v in the ordering. For example, [5, 7, 11, 2, 3, 8, 9, 10] is one of the topological sorts of the following directed graph.



An algorithm for computing a topological sort finds many applications. For instance, the vertices of the graph may represent a project consisting of a number of tasks, and the edges may represent constraints that one task must be done before another, where each node is a task; in this application, a topological ordering is just a valid execution sequence of the project. A directed graph has at least one topological sort if and only if the graph has no directed cycles, i.e. the graph is a directed acyclic graph (DAG). The following algorithm can be used to compute a topological sort for a directed graph if it is acyclic and output a message indicating that the graph is not acyclic.

```
 L ← Empty list, which will contain the sorted elements
S ← Set of all nodes with no incoming edge

while S is not empty do
    remove a node n from S
    add n to tail of L
    for each node m with an edge e from n to m do
        remove edge e from the graph
        if m has no other incoming edges then
            insert m into S

if graph has edges then
    return   "graph has at least one cycle"  in the output file
else
    return L (a topologically sort) in the output file
```

Implement the above algorithm in Python. Name your program "Q2.py".

The input file is a plain text file named "graph.txt", where all directed edges are listed one by one. In each row, only one directed edge (u, v) is given in the format "u,v", where u and v are strings representing nodes. You may assume that the node strings don't contain ','. The output file is called "topological_sort.txt".

Q3 (30 Marks) Write a program that takes two plain txt files named "fileA.txt" and "fileB.txt" and compute all words that have more occurrences in fileA.txt than fileB.txt and summarize the word occurrence results in an excel file named "WordOccurence.xlsx" in the following format:

- The output excel file contains three columns titled “Word”, “#(Occurrences) in A” and “#(Occurrences) in B”, respectively.
- Draw a barchart using the data in the last two columns, where the two series are labeled with A and B, respectively. The barchart graph should contain the occurrences of the top 5 most frequent words in fileA.txt. You should place the figure at the position (row 2, col G).

You are allowed to use functions in openpyxl. To learn openpyxl library, watch YuTube Video: https://www.youtube.com/watch?v=7YS6YDQKFh0

Text processing is never a simple task. The input files may contain 26 English letters together with 12 punctuations. For simplicity, we assume:

- There are not dashes, hyphens, underscores and quotations in the input files.
- Cases are ignored. For example, "My" and "my" are considered as the same word.
- No long word is broken across a line-break by a hyphen.

Name your program as “Q3.py”.