COSC 2P03 – Advanced Data Structures
Fall 2020
Assignment #2

Due Date: October 9th, 6pm **No late assignments will be accepted.**
This assignment accounts for 12% of your final grade and is worth a total of 120 marks

The goal of this assignment is to practice using general trees. You will build a simplified tree of virus and implement specified operations on it.

The human history is full of stories of combatting against pathogens (infectious agent) such as viruses, bacteria, fungi, etc. Among them, viruses are the smallest agents that often cause biggest health crises, for instance, what we are experiencing now is caused by SARS-CoV-2, a strain of the SARS-CoV virus under the Coronaviridae family. Based on the genetic material of a virus, it can be classified as either RNA virus (e.g. West Nile virus) or DNA virus (e.g. Smallpox virus). The following plot shows a simplified tree of virus.

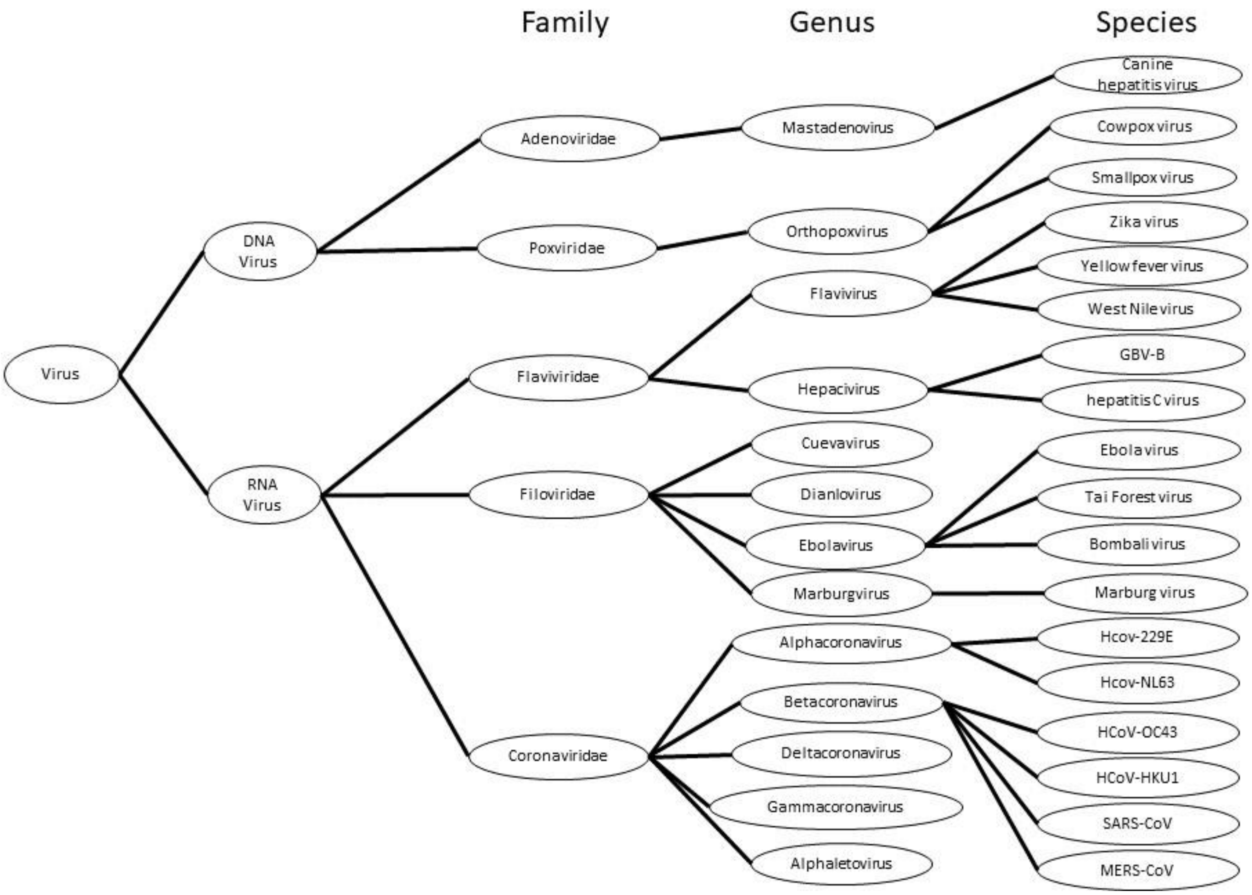

Figure 1. A simplified tree of virus. (The tree may be rotated for a better view.)

In this assignment, you are expected to complete the following tasks.

1. Define a class named `VirusTree` in Java. The data of this class is any virus tree structure, as shown in Figure 1. The information in each node of this general tree is the name of the virus as text shown in Figure 1. In addition to some necessary methods, the following methods must be implemented.
2. Define a method to create the virus tree using the representation of a general tree given in the class. This method should read the provided input file named "`tree_of_virus_input.txt`" which is a comma separated text file with the following format in each row:
   `Parent,Child1,Child2,…,Childk`
   Note that, in the input text file, this relationship can also be equivalently given in multiple rows as
   `Parent,Child1`
   `Parent,Child2`

…

`Parent,Childk`

At each time a row is processed and corresponding nodes are added, the function should print out the names of the parent node and its current children nodes in the format: “`Parent: Child1 -> Child2 -> … -> Childk`”. (30 marks)

3. Define a recursive method in the VirusTree class to return the height of the tree data. For example, the height of the tree in Figure 1 is 4. Suppose there are *n* nodes in the tree. You should print out the height of the tree structure. What is the time complexity of your method? (30 marks)
4. Define three methods respectively for breadth-first, pre-order and post-order tree traversals. Each method prints out the names of visited nodes in corresponding order. (30 marks)
5. Define a method to find the distance between two virus species. The distance between two virus species is defined as the length of any of them to their common ancestor node. For example, the distance between “Ebola virus” and “Bombali virus” should be 1. The distance between “Ebola virus” and “Marburg virus” should be 2. In your test, you should find (1) the distance between “HCoV-OC43” (causing common cold) and “Hcov-229E” (also causing common cold), (2) the distance between “SARS-CoV” and “Zika virus”. You should print out the information in this format: “`The distance between XXX and YYY is Z. They have common ancestor AAA`”. (30 marks)

Submission Requirements:

1. All submissions will be via Sakai. All of the following must be put in a single, zipped folder for submission:
2. All program files required to run your program. All code must be commented and properly documented.
3. A “printout” of your program, which has been printed into PDF format. Ensure that the first page of this PDF document contains a statement specifying whether you used IntelliJ (recommended), Eclipse or DrJava.
4. A printout (in PDF) of the output for your program. At the end, you should answer the question “What is the time complexity of your method?” raised in Task 3. Explain why. No marks will be given without explanation.