BIOS 6642 – Take-Home Assignment Four

Due at 11:59PM (Mountain Time), Wednesday, April 28, 2021

Please submit your source codes (.py or .ipynb file) and screenshots of the output of your codes. Your codes should be properly documented or commented.

- Q1 (30%). The *h*-index for a research scientist is defined as the maximum value of *h* such that he/she has published *h* papers that have each been cited at least *h* times. For example, if a scientist has published *5* papers with citations *[2,0,3,8,6]*, then his/her *h*-index is 3. Please write a program to calculate the *h*-index of each scientist defined in a file, ***citations.csv***, which contains the citation of each paper published by each scientist. Please download the file onto your computer and do not modify the content of the file. Your program needs to (1) read data from the file, and calculate and print the *h*-index for each scientist on the screen and (2) be able to properly handle the exception that the file does not exist.

- Q2 (35%). Please write a function that takes as input a non-empty list of integers, e.g., ***lst***, and returns the length of a subsequence of the list such that: (1) the elements of this subsequence appear in increasing order within the original sequence/list, and (2) the length of this subsequence is maximum. A subsequence is a sequence that can be derived from another sequence by deleting some or no elements without changing the order of the remaining elements. For example, *[1,3]* and *[2,3,4]* are subsequences of the sequence/list *[1,2,3,4,5]*. Assume that the length of input list can be any number between *1* and *100* (including *1* and *100*).

    For example,

    - If ***lst*** = *[9,3,8,6,7,1]*, then the function should print *3*, because the subsequence satisfying the two conditions above is *[3,6,7]*, which has length of *3*.
    - If ***lst*** = *[10,2,2,8,20,12,32,16,19]*, then the function should print *5*, because the subsequence satisfying the two conditions above is *[2,8,12,16,19]*, which has length of *5*.

    In addition to the function, please write some code to test the function.

- Q3 (35%). Please write a program to define a subclass, namely *Dog*, which inherits from the class, *Animal*, which is shown as follows. Each Dog object has a name (e.g., a string) and an age (e.g., an integer). Each Dog object also has parents, i.e., father dog and mother dog; the father and mother dogs of a Dog object are *None* if its parents are not specified.

```
class Animal:
    def __init__(self, name, age):
        self.name = name # a string
        self.age = age # an integer
    def get_age(self):
        return self.age
    def get_name(self):
        return self.name
```

The *Dog* subclass needs to

- Define or override an *__init__* method with proper parameters to initialize necessary attributes.

- Define or override a method such that applying the + operator on two Dog objects, *dog_a* + *dog_b*, produces another Dog object, *dog_c*, which will have an age 0. Additionally, the name of *dog_c* will be a string concatenating the names of its father and mother dogs; *dog_a* and *dog_b* are the parents of *dog_c*.

- Define or override a method to return the parents (i.e., the father and mother dogs) of a Dog object. If the parents of this Dog object are not specified, the method returns *None* for the father dog and *None* for the mother dog.

- Define or override a method such that **print(dog_a)** prints a string representation of the *dog_a* object as: *Dog:name:age,* where *name* and *age* should be replaced by the name and the age of *dog_a,* respectively.

Please do not modify the codes for the *Animal* class. In addition to the *Dog* subclass, please write some client code to test your subclass.