Faculty of Mathematical Economics

Data Structures and Algorithms

Instructor: **Nguyen Thanh Tuan**
DSEB Class of 2021 - 2024

Homework Assignment Week 5

Topic: Stack
Date Created: February 16, 2023

Problem 1: Constructing A Stack

Implement the `Stack` class that you learned in the class again with below structure:

```
class Stack:
    """LIFO Stack implementation using a Python list as underlying
   storage."""

    def __init__(self):
        """Create an empty stack"""
        pass

    def __len__(self):
        """Return the number of elements in the stack"""
        pass

    def is_empty(self):
        """Return True if the stack is empty"""
        pass

    def __str__(self):
        """Return string representation of the Stack"""
        pass

    def push(self, a):
        """Add element a to the top of the stack."""
        pass

    def top(self):
        """Return (but do not remove) the element at the top of
        the stack."""
        pass

    def pop(self):
        """Remove and return the element from the top of the stack."""
        pass
```

Using it to do the following tasks:

- Create a stack A with following elements: 4, 0, 1, 5, 9, 8, 2 (push 4 first).
- Move all the elements from stack A to an empty stack B without changing the order.
- Build a function to sort the stack B in ascending order.

Problem 2: Converting numeral system

Using `Stack` to:

- Convert a decimal number to a binary number.
- Convert a ternary number to a binary number. (You do not have to convert directly but try it if you can).

Ternary numeral system

Problem 3: Help Classmates

Mr. Tuan wants his students to help each other in the DSA class: every student should help out a classmate who scored less marks than him in DSA and whose ID number appears after him. But students are lazy and they do not want to search too far. They each pick the first ID number after them that fits the criteria. Note that one student may be selected by more than one classmate.

Implement a function using `Stack` to return a list of marks of the classmate that each student picks. If this student do not pick anyone, the mark is -1.

See the examples below:

```
marks = [3, 8, 5, 2, 9]
help_classmate(marks)
>>> [2, 5, 2, -1, -1]
 """Explanation:
 1. ID 1 has 3 marks. The first person who has less marks than him is
    ID 4, who has 2 marks.
 2. ID 2 has 8 marks, he helps student with 5 marks.
 3. ID 3 has 5 marks, he helps student 2 marks.
 4. ID 4 and 5 can not pick anyone as no student with higher ID has
    less marks than them.
 """

marks = [1, 2, 3, 4]
help_classmate(marks)
>>> [-1, -1, -1, -1]
 """Explanation: The marks are in increasing order. None of the
    students can find a classmate who has a higher ID and less marks
    than them."""
```

Guidelines for submission

- Your submission must be under the `.ipynb` format.
- Your submission will be graded and it is likely that homework grade will contribute as a component in your GPA.
- If your submission is later than the due date without special consideration approval, you will receive a penalty on your mark.