



Faculty of Mathematical Economics

Data Structures and Algorithms

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

Homework Assignment Week 4

Topic: Array-Based Sequences
Date Created: February 10, 2023

Problem 1: Constructing A Dynamic Array

Implement a **LeaderBoard** class that store high score information of players in a game. A leader board is limited to certain number of high scores that can be saved. Once that limit is reached, a new score only qualifies for the leader board if it is strictly higher than one of scores in the board.

For example, a leader board with capacity 10 is called a top 10 leader board which will contain 10 highest scores. Initially, it accepts first 10 ten scores in descending order, i.e scores from highest to lowest. When a later entry is considered:

- if it is lower than all of scores in top 10, then there will be no change in the lead board.
- if the entry is higher than one of top 10, it will be placed in appropriate rank and the last element of the leader board, corresponding to the current lowest score, will be removed.

Your **LeaderBoard** must follow the following structure:

```
1 class LeaderBoard:
2     """Fixed length sequence of high scores in descending order"""
3
4     def __init__(self, capacity):
5         """Initialize leader board with given maximum capacity
6         All entries are initially None
7         Capacity must be a positive integer"""
8         pass
9
10    def __len__(self):
11        """Return the number of non None elements in the board"""
12        pass
13
14    def get_player(self, player):
15        """Find a player in leader board using his name
```

```

16         Print not found if that player isn't in the board
17         """
18         pass
19
20     def __str__(self):
21         """Return string representation of the LeaderBoard"""
22         pass
23
24     def __delitem__(self, k):
25         """Delete score of rank k
26         k must be a positive interger and smaller than LEN of current
27         leader board """
28         pass
29
30     def add(self, new_entry):
31         """Consider adding new entry to high scores
32         new_entry: a tuple including score and player information"""
33         pass

```

Create a leader board for top 7 players in DSA Game. First 7 players are:

```

1 [ ('Catto', 25), ('Doggo', 28), ('Bunny', 19), ('Panda', 20), ('Snaky',
   30), ('Racoon', 23), ('Larma', 21)]

```

Do the following tasks:

- Add Owl with score 26 to the leader board.
- Add Piggy with score 20 to the leader board.
- Racoon was found cheated. Find and remove him from the leader board. Then Piggy is given the slot.
- Print out the leader board using the implemented string representation method.

Problem 2: Insertion Sort Algorithm

Implement Insertion Sort Algorithm to sort a list in ascending order.

[Insertion Sort Explanation](#)

Then, perform insertion sort on these arrays:

```

1 [6, 9, 10, 2, 0, -1, 4, 5, 8]
2 ['f', 'a', 'c', 'g', 'u', 'b', 'w', 'x', 'e']

```

Problem 3: Dynamic Programming

Implement a Fibonacci calculation method in the `DynamicArray` class that you have implemented on class. The method must be **non recursive**. What is time complexity of your solution?

Initialize an dynamic array object and use the implemented fibonacci method to calculate the 50th, 70th and 100th number of Fibonacci sequence.

Problem 4: Best Time To Buy And Sell Stock

You are given an array `price` where `price[i]` is the price of a given stock on the i^{th} day.

You want to maximize your profit by choosing a **single day** to buy one stock and choosing a **different day in the future** to sell that stock. Note that in the array you only choose one day to buy and one day to sell.

Implement a function to return the maximum profit you can achieve from this transaction, If you cannot achieve any profit, return 0. See the examples below:

```

1 price = [7, 1, 5, 3, 6, 4]
2 best_profit(price)
3 >>> 5
4 """Explanation: Buy on day 2 (price = 1) and sell on day 5 (price = 6)
   then profit = 6 - 1 = 5. Note that buying on day 2 and selling on
   day 1 is not allowed because you must buy before you sell."""
5
6 price = [7, 6, 4, 3, 1]
7 best_profit(price)
8 >>> 0
9 """Explanation: In the future prices keep decrease so your max profit
   is 0"""

```

Problem 5: Caesar Cipher

Read the information about Caesar Cipher on the internet and build a `CaesarCipher` class which contains two methods `encode` and `decode`.

- `encode` method takes in a string and return the encoding string in Caesar cipher.
- `decode` methods takes in a Caesar cipher string and return the true string.
- Both methods only work on alphabet letters. You can skip digits and special characters.

Perform Caesar encryption for the following strings with shift parameter = 4:

```
1 ['Hello Data Science', 'Lecture 4 Array']
```

Perform Caesar decryption for the following strings with shift parameter = 7:

```
1 ['Jhlzhy Jpwoly h zptwsl tlaovk', 'Jvunyhabshapvu, fvb mpuhssf mpupzolk
   aopz ovtldvyr']
```

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.