Assignment 4

CS221: Data Structures and Algorithms

Usman Institute of Technology

Fall 2019

Release Date: 24 November 2019

Submission Date: 30 November 2019 08:00 AM

- How to submit:
  - Create an account on http://www.turnitin.com/ as a Student
  - Use following information at time of sign-up

CS Section A

- Class ID: 22664649
- Enrollment Key: DSFALL19CSA

CS Section B

- Class ID: 22664651
- Enrollment Key: DSFALL19CSB

- **You have to submit single .py file. The name of your submitted file must be your roll number.**
- **Make sure that function names must be similar as asked in the assignment.**
- **You must read Academic Integrity at the end of this document.**

Note: This is a similar to Assignment 1 except this time the implementation is required using Circular Linked List instead of Arrays.

1. Create a class DynamicSharpSearch and write functions in Python whose parameters and return value are given below. The class uses Circular Linked List to store the data (not Python list)

   a. Create a constructor which takes a list as a parameter and store all elements in the list such that order of element must be preserved i.e. the first element in the given list must be at the first position of the linked list.

```
    def __init__(self, data):
        // your code goes here

Example: The following is the output if we execute the below given code
s = DynamicSharpSearch([5,4,7,1,4,9])
```

   b. Write a function Get which takes a parameter x and returns the element at x^{th} location.

```
    def Get(self, x):
        // your code goes here

Example: The following is the output if we execute the below given code
s = DynamicSharpSearch([5,4,7,1,4,9])
k = s.Get(0)
print(k)

Output:
5
```

c. Write a function **InsertAt** which takes two parameters **x** and **value**, and inserts a new element at x^{th} location.

```
    def InsertAt(self, x, value):
        // your code goes here

Example: The following is the output if we execute the below given code
s = DynamicSharpSearch([5,4,7,1,4,9])
s.InsertAt(0, 10)
k = s.Get(0)
print(k)

Output:
10
```

d. Write a function **SearchFirst** which takes a parameter ***value*** and returns the index of the location of the first occurrence of the value

```
    def SearchFirst(n):
        // your code goes here

Example: The following is the output if we execute the below given code
SharpSearch s = SharpSearch([5,4,7,1,4,9])
Print(s.SearchFirst(4))
Output:
1
```

e. Write a function **SearchLast** which takes a parameter ***value*** and returns the index of the location of the last occurrence of the value

```
    def SearchLast(n):
        // your code goes here

Example: The following is the output if we execute the below given code
SharpSearch s = SharpSearch([5,4,7,1,4,9])
Print(s.SearchLast(4))
Print(s.SearchLast(5))
Print(s.SearchLast(15))

Output:
4
0
-1
```

f. Write a function **Search** which takes a parameter ***value*** and returns the index of the location of the value. If this function calls for the first time, then it must return the location of the first occurrence of the value. On next call, it returns the next occurrence of the location or -1 if the element not found.

```
    def Search(n):
        // your code goes here
Example: The following is the output if we execute the below given code
SharpSearch s = SharpSearch([5,4,7,1,4,9])
Print(s.SearchFirst(4))
Print(s.SearchFirst(4))
Print(s.SearchFirst(4))
Print(s.SearchFirst(4))

Output:
1
4
-1
1
```

Academic Integrity

Each student in this course is expected to make sure that any work submitted by a student in this course for academic credit will be the **student's own work**. Scholastic dishonesty shall be considered a serious violation of these rules and regulations and is subject to strict disciplinary action. Scholastic dishonesty includes, but is not limited to, cheating on exams, plagiarism on assignments, and collusion.

PLAGIARISM: Plagiarism is the act of taking the work created by another person or entity and presenting it as one's own for the purpose of personal gain or of obtaining academic credit. Plagiarism includes the submission of or incorporation of the work of others without acknowledging its provenance or giving due credit according to established academic practices. This includes the submission of material that has been appropriated, bought, received as a gift, downloaded, or obtained by any other means. Students must not, unless they have been granted permission from all faculty members concerned, submit the same assignment or project for academic credit for different courses.

CHEATING: The term cheating shall refer to the use of or obtaining of unauthorized information in order to obtain personal benefit or academic credit.

COLLUSION: Collusion is the act of providing unauthorized assistance to one or more person or of not taking the appropriate precautions against doing so. Any student caught violating academic integrity will suffer an academic penalty. All violations of academic integrity will also be immediately reported to the Disciplinary Committee. Any student violating academic integrity a second time in this course will receive a failing grade for the course, and additional disciplinary sanctions may be administered through the Disciplinary Committee.

Conclusively, each student need to be take care of:

1. You must not share your solutions with other students. You are encouraged to discuss the problems but each student is supposed to take care of his or her own solution.
2. You must not submit solution of other students as yours.
3. You must duly cite all resources you used in development of your solution.