Middle East Technical University   Department of Computer Engineering

CENG 310

ALGORITHMS AND DATA STRUCTURES WITH PYTHON

Spring 2022-2023

Homework 4 - Sorting with Linked Lists

Due Date: 6/06/2023 23:59

Instructions. In this assignment you will implement bubblesort algorithm by using linked list structure.

1 Implementation of Linked List

A template file *hw4_template.py* will be provided to you in which you can find the skeleton code regarding Linked List. In the template you can find a class named LinkedList, an inner class named Node and each with their own member functions and variables. LinkedList class contains:

- __init__(self) function where it initializes LinkedList class.
- insert_last(self, e) is a function that adds a node with element e to the end of LinkedList class.
- print_contents(self) is a function that prints the current data of the nodes of the Linked List. Do not change anything in this function.
- head is a link that refers to the head node of the Linked List.
- size denotes the node count of the Linked List.

Node class is a simple class with only __init__(self, element, next) function with two member variables which are called data and next respectively.

Do not change the code related to Linked List class.

2 Programming Task

In this task, you will implement two functions named bubbleSinglePass(llist) and bubbleSort(llist) both of which take a linked list and apply either a single pass of bubblesort or the complete version of bubblesort. The input linked list will contain integers (possibly with duplicates) and your algorithms will adjust linked list nodes such a way that after calling the functions your functions will perform either a single pass or complete pass of the algorithm.

```
Sample I/O
Single Pass
linked.insert_last(3)
linked.insert_last(5)
linked.insert_last(1)
linked.insert_last(7)
linked.insert_last(5)
linked.insert_last(10)

linked.print_contents()

res = bubbleSinglePass(linked)
res.print_contents()

Output
3 -> 5 -> 1 -> 7 -> 5 -> 10
3 -> 1 -> 5 -> 5 -> 7 -> 10

Full Pass
linked.insert_last(3)
linked.insert_last(5)
linked.insert_last(1)
linked.insert_last(7)
linked.insert_last(5)
linked.insert_last(10)
linked.print_contents()

res = bubbleSort(linked)
res.print_contents()
Output
3 -> 5 -> 1 -> 7 -> 5 -> 10
1 -> 3 -> 5 -> 5 -> 7 -> 10
```

3 Submission & Grading

- Do not change the function and class definitions. You can not change the Singly Linked List structure.
- **Do not** leave your codes with unnecessary function codes and print statements. We will import your code and then call necessary functions ourselves to get the outputs. Any code that does not follow this rule will get penalized.
- You need to submit a single file with a *.py* extension.
- Be sure that you use the exact names for the classes and functions that you will implement. You can implement other helper functions to use inside of the functions you are provided.
- You should not worry about the run time of your program as long as it gives the desired output. However, extraordinarily long (in terms of time) solutions may result in getting no credit from particular test case.

- **Do not** try to change data of linked nodes to make them in sorted order. You **must** change the order of nodes by changing node references. Any solution that does not reorder nodes and simply changes the data variable of Node Class will get 0.