

# CSC1001: Introduction to Computer Science

## Programming Methodology

### Assignment 4

#### Assignment description:

This assignment will be worth **16%** of the final grade.

You should write your code for each question in a **.py** file (please name it using the question name, e.g. **q1.py**). Please pack all your **.py** files into a single **.zip** file, name it using your **student ID** (e.g. if your student ID is 123456, then the file should be named as 123456.zip), and then submit the **.zip** file via Blackboard.

Please also write a text file, which provide the details. (Note that the report should be submitted as PDF) The report should be included in the **.zip** file as well.

Please note that, the teaching assistant may ask you to **explain the meaning of your program**, to ensure that the codes are indeed written by yourself. **Plagiarism will not be tolerated**. We may check your code using Blackboard.

This assignment is due on **5:00PM, 21 May (Friday)**. For **each day** of late submission, you will lose **10%** of your mark in this assignment. If you submit **more than three days** later than the deadline, you will receive **zero** in this assignment.

#### Question 1 (**20%** of this assignment):

Write a Python class called **SinglyLinkedList**. The class should contain a method named **recursive\_count** which **recursively** counts the number of nodes in a **singly linked list**. The input of the **recursive\_count** function should be a **reference** pointing to the first node of the linked list. The output of the function should be the **number of nodes** in that linked list.

Note that you also need to implement the constructor and **insert** method (which takes data to be inserted as argument) for **SinglyLinkedList** class. You may also implement any other method within the class as you want. Your class should look like the following. (Please follow the same names of class and methods in the following example)

```
class Node:
    def __init__(self, element, pointer):
        self.element = element
        self.pointer = pointer

class SinglyLinkedList:
    def __init__(self):
        pass # constructor to be implemented by yourself

    def insert(self, data):
        pass # insert method to be implemented by yourself

    def recursive_count(self, node):
        pass # recursive count to be implemented by yourself
```

**Question 2 (30% of this assignment):**

Follow the same setting in Q1. Write the method named `quick_sort` inside `SinglyLinkedList` class which uses quick sort algorithm to sort over a `singly linked list`.

The input of your function should be a `reference` pointing to the first node of a linked list, and the output of your function should also be a `reference` to the first node of a linked list, in which the data have been sorted into the `ascending order`.

You may use the `LinkedQueue` class we introduced in the lecture directly in your program. Note that you also need to implement the constructor and `insert` method (which takes data to be inserted as argument) for `SinglyLinkedList` class.

You may also implement any other method within the class as you want. Your class should look like the following. (Please follow the same names of class and methods in the following example)

```
class Node:
    def __init__(self, element, pointer):
        self.element = element
        self.pointer = pointer

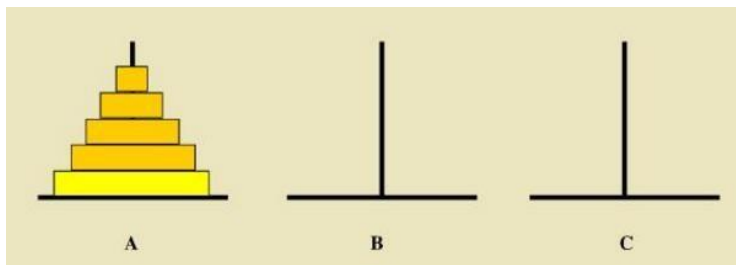
class SinglyLinkedList:
    def __init__(self):
        pass # constructor to be implemented by yourself

    def insert(self, data):
        pass # insert method to be implemented by yourself

    def quick_sort(self, node):
        pass # recursive count to be implemented by yourself
```

**Question 3 (50% of this assignment):**

The `Tower of Hanoi` is a mathematical game or puzzle. It consists of three rods, and a number of disks of different sizes which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape. The following figure shows the initial state of the Tower of Hanoi with 5 disks.



The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.

2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack, i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

Assume that initially all the disks are placed on rod A. Write a **non-recursive** Python function to print out the steps to move all the disks from rod A to rod C via rod B (Hint: a recursive algorithm can be converted into a non-recursive algorithm using stack). The header of the function is:

```
def HanoiTower(n)
```

Here **n** represents the number of disks. For example, when **n = 3** your function should output:

```
A --> C
A --> B
C --> B
A --> C
B --> A
B --> C
A --> C
```

You should write codes to prompt user to input **n** and output the results.