

CSC1001: Introduction to Computer Science

Programming Methodology

Assignment 4

Assignment description:

This assignment will be worth **16%** of the final grade. **Please follow the template provided on Blackboard to write your code. Otherwise, you will get zero points.**

You should write your code for each question in a **.py** file (please name it using the question name, e.g. for Question 1, name it **q1.py**). Please pack all your **.py** files into a single **.zip** file following the provided example 123456.zip on Blackboard, 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 about how to run your code for each question. The text file 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. We may check **whether your program is too similar to your fellow students' code** using Blackboard.

This assignment is due on **5:00PM, 26 Dec. (Tue)**. 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 function to implement a **recursive** algorithm which counts the number of nodes in a **singly linked list**. The input of the 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.

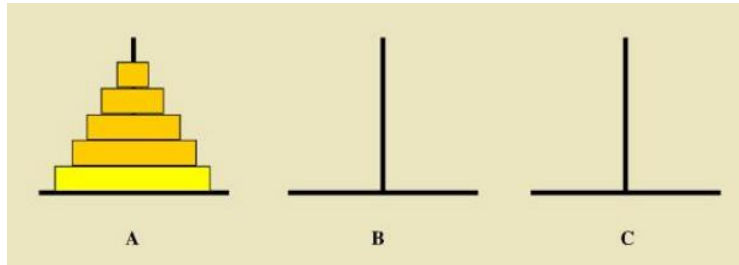
Question 2 (**30%** of this assignment):

Write a Python function to implement the quick sort algorithm over a **singly linked list**. The input of your function should be a **variable** pointing to the first node of a linked list, and the output of your function should also be a **variable** pointing to the first node of a linked list, in which the data have been sorted into the **ascending order**. You may use the **LinkedList** class we introduced in the lecture directly in your program.

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** the steps should be:

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